

ROBERTA SUERO

**OTIMIZAÇÃO DE PARÂMETROS DO MÉTODO *MULTIGRID* ALGÉBRICO PARA
PROBLEMAS DIFUSIVOS BIDIMENSIONAIS**

Tese apresentada como requisito parcial para a
obtenção do título de doutor em Ciências no
Programa de Pós-Graduação em Métodos Numéricos
em Engenharia, Setores de Tecnologia e Ciências
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Carlos Henrique Marchi.

Co-orientador: Prof. Dr. Marcio Augusto Villela
Pinto.

CURITIBA

2010

S944o

Suero, Roberta

Otimização de parâmetros do método multigrid algébrico para problemas difusivos bidimensionais [manuscrito] / Roberta Suero. – Curitiba, 2010. 167f. : il. [algumas color.] ; 30 cm.

Impresso.

Tese (doutorado) - Universidade Federal do Paraná, Setor de Tecnologia e Ciências Exatas, Programa de Pós-graduação em Métodos Numéricos em Engenharia, 2010.

Orientador: Carlos Henrique Marchi -- Co-orientador: Marcio Augusto Villela Pinto.

1. Dinâmica dos fluidos – Processamento de dados. 2. Análise numérica (Multigrid). I. Universidade Federal do Paraná. II. Marchi, Carlos Henrique. III. Pinto, Marcio Augusto Villela. IV. Título.

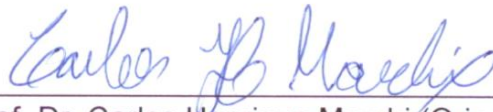
CDD: 532.05

TERMO DE APROVAÇÃO

ROBERTA SUERO

OTIMIZAÇÃO DE PARÂMETROS DO MÉTODO *MULTIGRID* ALGÉBRICO PARA PROBLEMAS DIFUSIVOS BIDIMENSIONAIS

Tese aprovada como requisito parcial para obtenção do grau de Doutor no Programa de Pós-Graduação em Métodos Numéricos em Engenharia – PPGMNE, Setor de tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:



Prof. Dr. Carlos Henrique Marchi (Orientador)

Programa de Pós-Graduação em Métodos Numéricos em Engenharia – PPGMNE, UFPR



Prof. Dr. Marcio Augusto Villela Pinto (Co-orientador)

Departamento de Engenharia Mecânica – UFPR



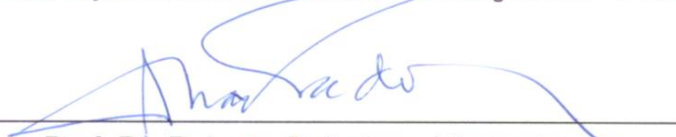
Prof. Dr. Leandro Franco de Souza

Universidade de São Paulo – USP



Prof. Dr. Luciano Kiyoshi Araki

Programa de Pós-Graduação em Métodos Numéricos em Engenharia – PPGMNE, UFPR



Prof. Dr. Roberto Dalledone Machado

Programa de Pós-Graduação em Métodos Numéricos em Engenharia – PPGMNE, UFPR

Curitiba, 10 de junho de 2010

À minha família pelo amor e apoio incondicionais.

AGRADECIMENTOS

Agradeço ao Prof. Carlos Henrique Marchi por aceitar me orientar neste trabalho, agradeço também pelo apoio e os conhecimentos que me foram transmitidos até o momento. Ao meu co-orientador Prof. Marcio Augusto Villela Pinto pela orientação, incentivo e amizade.

Agradeço aos amigos que, direta ou indiretamente, me ajudaram no desenvolvimento deste trabalho, seja com ensinamentos ou em momentos de descontração, em especial aos amigos do LENA.

Aos professores, colegas e funcionários do CESEC, em especial à Maristela Bandil, pela amizade e dedicação.

Gostaria de agradecer também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento deste trabalho.

Agradeço ainda aos membros da banca, Prof. Dr. Leandro Franco Souza, Prof. Dr. Luciano Kiyoshi Araki e Prof. Dr. Roberto Dalledone Machado pelo tempo dispensado, sugestões e críticas apresentadas a esta tese.

RESUMO

Este trabalho apresenta comparações de parâmetros entre os métodos *multigrid* algébrico (AMG) e *multigrid* geométrico (GMG) para as equações bidimensionais de Laplace e Poisson, em malhas estruturadas quadrangulares e triangulares. Os parâmetros analisados são: número de iterações internas no *solver*, número de malhas e número de incógnitas. Para o AMG, também são estudados os efeitos do fator de redução de malha e do fator de forte dependência na malha grossa sobre o tempo de CPU necessário para obter a solução numérica. Para malhas quadrangulares é empregado o método de diferenças finitas, e para malhas triangulares, o de volumes finitos. Os resultados são obtidos com uma adaptação do código computacional AMG1R6 de Ruge e Stüben. Para o AMG são usadas as seguintes componentes *multigrid*: restrição por engrossamento padrão, prolongação padrão, esquema de correção (CS), *solver* Gauss-Seidel lexicográfico e ciclo V. São feitos estudos comparativos entre os tempos de CPU do método *multigrid* geométrico, *multigrid* algébrico e *singlegrid* (método de malha única). Verificou-se que: 1) o número ótimo de iterações internas obtidas para o AMG e GMG, em malhas quadrangulares, é o mesmo, porém diferente para malhas triangulares; 2) o número ótimo de malhas é o número máximo, tanto para malhas quadrangulares quanto para malhas triangulares; 3) o AMG mostrou-se sensível à variação do fator de redução de malha e do fator de forte dependência na malha grossa, tanto com relação às equações abordadas, quanto aos tipos de malha e 4) para malhas quadrangulares, o GMG resolve o problema em 20% do tempo gasto pelo AMG.

Palavras-chave: dinâmica dos fluidos computacional, *multigrid* algébrico, *multigrid* geométrico, volumes finitos, diferenças finitas.

ABSTRACT

This work presents comparisons of parameters between the algebraic multigrid (AMG) and geometric multigrid (GMG) methods for Laplace and Poisson two-dimensional equations in square and triangular structured grids. The analyzed parameters are: the number of inner iterations in the solver, the number of grids and number of unknowns. For AMG, the effects of the grid reduction factor and the strong dependence factor in the coarse grid on the necessary CPU time to obtain the numeric solution are studied. For square grids the finite difference method is used, and for the triangular grids, the finite volume one. The results are obtained with the use of an adapted computational code from the original AMG1R6 of Ruge and Stüben. For the AMG the following multigrid components are used: restriction by the standard coarsening, standard interpolation, correction scheme (CS), lexicographic Gauss-Seidel as solver and V cycle. Comparative studies among the CPU time of the geometric and algebraic multigrid methods and singlegrid (method of unique mesh) are made. It was verified that: 1) the optimum number of inner iterations obtained for AMG and GMG, in square grids, is the same, however it has a value different for triangular grids; 2) the optimum number of grids is the maximum number, for both square and triangular grids; 3) AMG was shown to be sensitive to both the variation of the grid reduction factor and the strong dependence factor in the coarse grid, in relation to the approached equations, as like to the mesh types, and 4) in square grids, the GMG solves the problem in 20% of the time spend for AMG.

Keywords: computational fluid dynamics, algebraic multigrid, geometric multigrid, finite volume, finite difference.

LISTA DE FIGURAS

Figura 1.1: Malha quadrangular	27
Figura 1.2: Malha triangular	27
Figura 2.1: Malha geométrica utilizada para Gauss-Seidel	40
Figura 2.2: Modos de Fourier	45
Figura 2.3: Comportamento da suavização do erro em métodos iterativos	46
Figura 2.4: Número de ondas $k=4$, sobre uma malha fina com $N=12$ e sobre uma malha grossa com $N=6$ (Briggs et al., 2000)	47
Figura 2.5: Seqüência do procedimento de seleção dos pontos (Adaptado de Falgout, 2006).....	54
Figura 2.6: Ilustração de influências fortes e fracas sobre o ponto i (BRIGGS et al., 2000).....	56
Figura 2.7: Ciclo V (Adaptado de Craig, 2007).....	60
Figura 2.8: Discretização com diferenças finitas	65
Figura 2.9: Malha estruturada - volume coincidente com o elemento (CORDAZZO, 2006)	67
Figura 2.10: Volume de controle para uma malha triangular	68
Figura 2.11: Condições de contorno com volumes fictícios (MALISKA, 2004)	71
Figura 3.1: Erros envolvidos nos métodos de engenharia (Marchi, 2001)	72
Figura 3.2: Ordens efetiva e aparente para temperatura no ponto médio para a equação de Laplace em malhas quadrangulares	77
Figura 3.3: Ordens efetiva e aparente para temperatura média para a equação de Laplace em malhas quadrangulares	77
Figura 3.4: Ordens efetiva e aparente para a norma l_{∞} do erro para a equação de Laplace em malhas quadrangulares	77
Figura 3.5: Ordens efetiva e aparente para a norma \bar{l}_1 do erro para a equação de Laplace em malhas quadrangulares	77
Figura 3.6: Erros numéricos para o problema linear em malhas quadrangulares	77
Figura 3.7: Erros numéricos para o problema senoidal em malhas quadrangulares	77
Figura 3.8: Comportamento da norma l_{∞} do erro para a equação de Laplace em malhas quadrangulares	78
Figura 3.9: Tempo de CPU(s) <i>versus</i> número de incógnitas para a equação de Laplace em malhas quadrangulares	80

Figura 3.10: Ordens efetiva e aparente para a temperatura no ponto médio para a equação de Laplace em malhas triangulares	82
Figura 3.11: Ordens efetiva e aparente para a temperatura média para a equação de Laplace em malhas triangulares	82
Figura 3.12: Ordens efetiva e aparente para a norma l_∞ do erro para a equação de Laplace em malhas triangulares	83
Figura 3.13: Ordens efetiva e aparente para a norma \bar{l}_1 do erro para a equação de Laplace em malhas triangulares	83
Figura 3.14: Erros numéricos para o problema linear em malhas triangulares.....	83
Figura 3.15: Erros numéricos para o problema senoidal em malhas triangulares	83
Figura 3.16: Comportamento da norma l_∞ do erro para a equação de Laplace em malhas triangulares.....	84
Figura 3.17: Tempo de CPU(s) <i>versus</i> número de incógnitas para a equação de Laplace em malhas triangulares.....	85
Figura 3.18: Ordem efetiva – equação de Poisson em malhas quadrangulares	87
Figura 3.19: Ordem aparente – equação de Poisson em malhas quadrangulares	87
Figura 3.20: Erros numéricos – equação de Poisson em malhas quadrangulares.....	87
Figura 3.21: Norma l_∞ do erro para a equação de Poisson em malhas quadrangulares.....	87
Figura 3.22: Tempo de CPU(s) <i>versus</i> número de incógnitas para a equação de Poisson em malhas quadrangulares	88
Figura 4.1: Tempo de CPU <i>versus</i> número de iterações internas para o problema linear em malhas quadrangulares	92
Figura 4.2: Tempo de CPU <i>versus</i> número de níveis para o problema linear em malhas quadrangulares	93
Figura 4.3: Influência do fator de redução de malha no tempo de CPU para o problema linear em malhas quadrangulares.....	95
Figura 4.4: Efeito do fator de forte dependência na malha grossa no tempo de CPU para o problema linear em malhas quadrangulares.....	97
Figura 4.5: Efeito do número de incógnitas no tempo de CPU para o problema linear em malhas quadrangulares	98
Figura 4.6: AMG otimizado <i>versus</i> AMG padrão para o problema linear em malhas quadrangulares	100

Figura 4.7: Tempo de CPU <i>versus</i> número de iterações internas para o problema senoidal em malhas quadrangulares	102
Figura 4.8: Tempo de CPU <i>versus</i> número de níveis para o problema senoidal em malhas quadrangulares	104
Figura 4.9: Influência do fator de redução de malha no tempo de CPU para o problema senoidal em malhas quadrangulares	105
Figura 4.10: Influência do fator de forte dependência na malha grossa no tempo de CPU para o problema senoidal em malhas quadrangulares.....	106
Figura 4.11: Efeito do número de incógnitas no tempo de CPU para o problema senoidal em malhas quadrangulares	108
Figura 4.12: AMG otimizado <i>versus</i> AMG padrão para o problema senoidal em malhas quadrangulares	110
Figura 4.13: Tempo de CPU <i>versus</i> número de iterações internas para a equação de Poisson em malhas quadrangulares	112
Figura 4.14: Tempo de CPU <i>versus</i> número de níveis para a equação de Poisson em malhas quadrangulares	114
Figura 4.15: Influência do fator de redução de malha no tempo de CPU para a equação de Poisson em malhas quadrangulares	115
Figura 4.16: Influência do fator de forte dependência na malha grossa no tempo de CPU para a equação de Poisson em malhas quadrangulares.....	116
Figura 4.17: Efeito do número de incógnitas no tempo de CPU para a equação de Poisson em malhas quadrangulares	118
Figura 4.18: AMG otimizado <i>versus</i> AMG padrão para a equação de Poisson em malhas quadrangulares	120
Figura 4.19: Número de iterações internas para malhas quadrangulares.....	122
Figura 4.20: Número de níveis para malhas quadrangulares	123
Figura 4.21: Fator de redução de malha para malhas quadrangulares	123
Figura 4.22: Fator de forte dependência na malha grossa para malhas quadrangulares	124
Figura 4.23: Tempo de CPU para AMG, GMG e SG para malhas quadrangulares	125
Figura 5.1: Tempo de CPU <i>versus</i> número de iterações internas para o problema linear em malhas triangulares.....	131

Figura 5.2: Tempo de CPU <i>versus</i> número de níveis para o problema linear em malhas triangulares	132
Figura 5.3: Influência do fator de redução de malha no tempo de CPU para o problema linear em malhas triangulares	134
Figura 5.4: Efeito do fator de forte dependência na malha grossa no tempo de CPU para o problema linear em malhas triangulares	136
Figura 5.5: Efeito do número de incógnitas no tempo de CPU para o problema linear em malhas triangulares.....	137
Figura 5.6: AMG otimizado <i>versus</i> AMG padrão para o problema linear em malhas triangulares	138
Figura 5.7: Tempo de CPU <i>versus</i> número de iterações internas para o problema senoidal em malhas triangulares	140
Figura 5.8: Tempo de CPU <i>versus</i> número de níveis para o problema senoidal em malhas triangulares.....	142
Figura 5.9: Influência do fator de redução de malha no tempo de CPU para o problema senoidal em malhas triangulares	144
Figura 5.10: Influência do fator de forte dependência na malha grossa no tempo de CPU para o problema senoidal em malhas triangulares	145
Figura 5.11: Efeito do número de incógnitas no tempo de CPU para o problema senoidal em malhas triangulares.....	147
Figura 5.12: AMG otimizado <i>versus</i> AMG padrão para o problema senoidal em malhas triangulares	148
Figura 5.13: Número de iterações internas para malhas triangulares	150
Figura 5.14: Número de níveis para malhas triangulares.....	151
Figura 5.15: Fator de redução de malha para malhas triangulares.....	151
Figura 5.16: Fator de forte dependência na malha grossa para malhas triangulares	152
Figura 5.17: Tempo de CPU para AMG, GMG e SG para malhas triangulares.....	153

LISTA DE TABELAS

Tabela 1.1: Comparação entre os métodos de solução de problemas em engenharia (Adaptada de Tannehill et al., 1997)	23
Tabela 1.2: Comparações entre <i>multigrid</i> geométrico e <i>multigrid</i> algébrico (Adaptado de CHANG et al., 1996).....	31
Tabela 2.1: Procedimento iterativo Gauss-Seidel (BURDEN e FAIRES, 2009)	41
Tabela 2.2: Algoritmo para fase <i>setup</i> do AMG (Adaptado de Cleary et al., 2000)	58
Tabela 2.3: Escolha preliminar do ponto em C (Ruge e Stüben, 1986).....	58
Tabela 2.4: Escolha final do ponto em C e definição dos pesos da interpolação (Adaptado de Iwamura et al., 2003)	58
Tabela 2.5: Algoritmo para gerar a matriz de coeficientes (HAASE e LANGER, 2002)	60
Tabela 2.6: Algoritmo CS para o AMG com duas malhas (Adaptado de BRIGGS et al., 2000)	61
Tabela 3.1: Comparativo da norma l_∞ do erro para a equação de Laplace em malhas quadrangulares	79
Tabela 3.2: Valores de c e p para AMG, GMG e SG para a equação de Laplace em malhas quadrangulares	81
Tabela 3.3: Comparativo da norma l_∞ do erro para a equação de Laplace em malhas triangulares	84
Tabela 3.4: Valores de c e p para AMG, GMG e SG para a equação de Laplace em malhas triangulares.....	85
Tabela 3.5: Comparativo da norma l_∞ do erro para a equação de Poisson	87
Tabela 3.6: Valores de c e p para AMG, GMG e SG para a equação de Poisson em malhas quadrangulares	89
Tabela 4.1: Diferenças entre os tempos de CPU para θ considerando o problema linear em malhas quadrangulares	96
Tabela 4.2: Diferenças entre os tempos de CPU para ε considerando o problema linear em malhas quadrangulares	97
Tabela 4.3: Valores de p e c para o problema linear em malhas quadrangulares	99
Tabela 4.4: Parâmetros estudados para o AMG para o problema linear em malhas quadrangulares	99
Tabela 4.5: Valores de p e c para o problema linear em malhas quadrangulares	100
Tabela 4.6: Número de ciclos para AMG e GMG para o problema linear em malhas quadrangulares	101

Tabela 4.7: Diferenças entre os tempos de CPU para θ considerando o problema senoidal em malhas quadrangulares	106
Tabela 4.8: Diferenças entre os tempos de CPU para ε considerando o problema senoidal em malhas quadrangulares	107
Tabela 4.9: Valores de p e c para o problema senoidal em malhas quadrangulares	108
Tabela 4.10: Parâmetros estudados para o AMG para o problema senoidal em malhas quadrangulares	109
Tabela 4.11: Valores de p e c para o problema senoidal em malhas quadrangulares	110
Tabela 4.12: Número de ciclos para AMG e GMG para o problema senoidal em malhas quadrangulares	111
Tabela 4.13: Diferenças entre os tempos de CPU para θ considerando a equação de Poisson em malhas quadrangulares	115
Tabela 4.14: Diferenças entre os tempos de CPU para ε considerando a equação de Poisson em malhas quadrangulares	117
Tabela 4.15: Valores de p e c para a equação de Poisson em malhas quadrangulares	119
Tabela 4.16: Parâmetros estudados para o AMG para a equação de Poisson em malhas quadrangulares	119
Tabela 4.17: Valores de p e c para a equação de Poisson em malhas quadrangulares	120
Tabela 4.18: Número de ciclos para AMG e GMG para a equação de Poisson em malhas quadrangulares	121
Tabela 4.19: Valores de p e c para malhas quadrangulares	125
Tabela 4.20: Ganhos obtidos com as otimizações para malhas quadrangulares.....	126
Tabela 4.21: Número de ciclos para AMG e GMG para malhas quadrangulares	126
Tabela 5.1: Número de níveis e de ciclos para o problema linear resolvido em malhas triangulares para diferentes valores de θ	133
Tabela 5.2: Diferenças entre os tempos de CPU para θ considerando o problema linear em malhas triangulares.....	134
Tabela 5.3: Diferenças entre os tempos de CPU para ε considerando o problema linear em malhas triangulares.....	136
Tabela 5.4: Valores de p e c para o problema linear em malhas triangulares.....	138
Tabela 5.5: Parâmetros estudados para o AMG para o problema linear em malhas triangulares ...	139

Tabela 5.6: Valores de p e c para AMG padrão e otimizado para o problema linear em malhas triangulares.....	139
Tabela 5.7: Diferenças entre os tempos de CPU para L considerando o problema senoidal em malhas triangulares	142
Tabela 5.8: Número de níveis e de ciclos para o problema senoidal resolvido em malhas triangulares para diferentes valores de θ	143
Tabela 5.9: Diferenças entre os tempos de CPU para θ considerando o problema senoidal em malhas triangulares	144
Tabela 5.10: Diferenças entre os tempos de CPU para ε considerando o problema senoidal em malhas triangulares	146
Tabela 5.11: Valores de p e c para o problema senoidal em malhas triangulares	147
Tabela 5.12: Parâmetros estudados para o AMG para o problema senoidal em malhas triangulares	148
Tabela 5.13: Valores de p e c para AMG padrão e otimizado para o problema senoidal em malhas triangulares.....	149
Tabela 5.14: Valores de p e c para malhas triangulares.....	153
Tabela 5.15: Ganhos obtidos com otimizações em malhas triangulares.....	154
Tabela B.0.1: Diferenças entre os tempos de CPU para o fator de redução de malha para o problema linear em malhas quadrangulares	168

LISTA DE SÍMBOLOS

a_{ij}	Coeficientes da matriz A
A	Matriz de coeficientes
c	Constante relacionada ao ajuste de curvas pelo método dos mínimos quadrados
C	Conjunto de variáveis que estão na malha grossa
C^i	Conjunto dos pontos em C que estão conectados a $i \in F$
C_i	Conjunto dos pontos vizinhos da malha grossa que influenciam fortemente i
D_x	Deslocamento na direção x
D_y	Deslocamento na direção y
D_i	Conjunto dos pontos vizinhos que não estão na malha grossa
D_i^s	Conjunto dos pontos vizinhos da malha fina que influenciam fortemente i
D_i^w	Conjunto dos pontos que influenciam fracamente i
e	Erro algébrico
\bar{e}^h	Erro suavizado
E	Erro de discretização
f	Vetor independente do sistema linear
F	Conjunto de variáveis que não estão na malha grossa
F^i	Conjunto dos pontos em F que estão conectados a $i \in F$
h	Malha fina
H	Malha grossa
I_h^H	Operador de restrição
I_H^h	Operador de prolongação
\vec{J}	Fluxo difusivo
k	Modos de Fourier
l_2	Norma Euclidiana
l_∞	Norma do máximo (ou norma infinito)

\bar{l}_1	Norma L_1 média para o erro numérico
L	Número de níveis
$L_{máximo}$	Número máximo de níveis
$L_{ótimo}$	Número ótimo de níveis
L^0	Norma dos resíduos na estimativa inicial (iteração 0)
L^m	Norma dos resíduos na iteração m
m	Contador do número de iterações para o método Gauss-Seidel
N	Número de incógnitas
N_i	Conjunto de pontos que determina a vizinhança de i
N_x	Quantidade de pontos na direção x
N_y	Quantidade de pontos na direção y
p	Expoente relacionado ao ajuste de curvas pelo método dos mínimos quadrados
p_E	Ordem efetiva do erro de discretização
p_L	Ordem assintótica do erro de discretização
p_U	Ordem aparente do erro de discretização
q	Razão de refino entre as malhas
r	Resíduo
r^m	Resíduo na iteração m
S_h	Suavizador
S_i	Conjunto de pontos que influencia fortemente i
S_i^D	Conjunto de pontos fortemente dependentes de C_i
S_i^T	Conjunto de pontos que dependem fortemente de i
T	Tempo de CPU
T_m	Temperatura média
U	Vetor de incógnitas
U	Conjunto de variáveis indefinidas
\bar{u}^h	Solução suavizada
V	Solução aproximada do sistema de equações
V_x	Velocidade na direção x

V_y	Velocidade na direção y
w_{ij}	Função de interpolação do <i>multigrid</i> algébrico

Letras Gregas

α	Difusividade térmica
Δx	Tamanho dos volumes de controle na direção x
Δy	Tamanho dos volumes de controle na direção y
ε	Fator de forte dependência na malha grossa
$\varepsilon_{\text{ótimo}}$	Valor ótimo para o fator de forte dependência na malha grossa
ν	Número de iterações internas
$\nu_{\text{ótimo}}$	Número ótimo de iterações internas
∇	Operador nabla
θ	Fator de redução da malha
$\theta_{\text{ótimo}}$	Valor ótimo para o fator de redução de malha
λ	Medida de importância
ϕ	Solução numérica da variável de interesse
Φ	Solução analítica da variável de interesse
Ω	Conjunto de índices
Γ^ϕ	Coefficiente de difusão

Subíndices

N	face localizada acima do ponto geral nodal
S	face localizada abaixo do ponto geral nodal
E	face localizada à direita do ponto geral nodal
W	face localizada à esquerda do ponto geral nodal
I	posição do nó na direção x
J	posição do nó na direção y
K	k -ésima iteração
H	malha fina
H	malha grossa

Abreviaturas e Siglas

ADI	<i>Alternate Direction Implicit Method</i>
AMG	<i>Algebraic Multigrid Method</i>
AMG1R6	Código computacional de autoria de Ruge e Stüben
CDS	<i>Central Differencing Scheme</i>
CFD	<i>Computational Fluid Dynamics</i>
CPU	<i>Central Processing Unit</i>
GMG	<i>Geometric Multigrid Method</i>
<i>m_maximo</i>	Número máximo de iterações para o método Gauss-Seidel
MDF	Método de diferenças finitas
MSI	<i>Modified Strongly Implicit Method</i>
MVF	Método dos volumes finitos
SG	Método <i>Singlegrid</i>
SOR	<i>Sucessive Over Relaxation</i>
TDMA	<i>TriDiagonal Matrix Algorithm</i>
<i>Tol</i>	Tolerância empregada para se interromper o processo iterativo

SUMÁRIO

1	Introdução	22
1.1	Generalidades em dinâmica dos fluidos computacional	22
1.2	O problema	26
1.3	Motivação	27
1.4	Revisão bibliográfica.....	30
1.5	Objetivos	36
1.6	Organização do texto.....	37
2	Fundamentação teórica	39
2.1	Métodos iterativos e análise de convergência	39
2.2	Equação residual.....	42
2.3	Método <i>multigrid</i> algébrico.....	44
2.3.1	Princípios fundamentais	44
2.3.2	Teoria básica do AMG	49
2.3.3	Engrossamento	52
2.3.4	Prolongação	54
2.3.5	Algoritmos	57
2.4	Modelos matemáticos e numéricos	61
2.4.1	Formulação para o método de diferenças finitas	65
2.4.2	Formulação para o método dos volumes finitos.....	66
3	Verificação do código computacional	72
3.1	Equação de Laplace – Malhas quadrangulares.....	76
3.2	Equação de Laplace – Malhas triangulares	82
3.3	Equação de Poisson – Malhas quadrangulares.....	86
4	Resultados para malhas estruturadas quadrangulares	90
4.1	Equação de Laplace – Problema linear	91
4.1.1	Número de iterações internas (ν)	91
4.1.2	Número de níveis (L)	92
4.1.3	Fator de redução de malha (θ)	94
4.1.4	Fator de forte dependência na malha grossa (ε)	96

4.1.5	Número de incógnitas (N)	98
4.1.6	Otimização do algoritmo para o AMG	99
4.1.7	Comparação entre número de ciclos para AMG e GMG	101
4.2	Equação de Laplace – Problema senoidal	101
4.2.1	Número de iterações internas (ν)	102
4.2.2	Número de níveis (L)	103
4.2.3	Fator de redução de malha (θ)	105
4.2.4	Fator de forte dependência na malha grossa (ε)	106
4.2.5	Número de incógnitas (N)	107
4.2.6	Otimização do algoritmo para o AMG	109
4.2.7	Comparação entre número de ciclos para AMG e GMG	110
4.3	Equação de Poisson	111
4.3.1	Número de iterações internas (ν)	111
4.3.2	Número de níveis (L)	113
4.3.3	Fator de redução de malha (θ)	114
4.3.4	Fator de forte dependência na malha grossa (ε)	116
4.3.5	Número de incógnitas (N)	117
4.3.6	Otimização do algoritmo para o AMG	119
4.3.7	Comparação entre número de ciclos para AMG e GMG	121
4.4	Comparativo para malhas quadrangulares	121
4.4.1	Número de iterações internas (ν)	122
4.4.2	Número de níveis (L)	122
4.4.3	Fator de redução de malha (θ)	123
4.4.4	Fator de forte dependência na malha grossa (ε)	124
4.4.5	Número de incógnitas (N)	125
4.4.6	Otimização do algoritmo para o AMG	126
4.4.7	Comparação entre número de ciclos para AMG e GMG	126
4.5	Conclusões do Capítulo 4	127
5	Resultados para malhas estruturadas triangulares	129
5.1	Equação de Laplace – Problema linear	130

5.1.1	Número de iterações internas (ν)	130
5.1.2	Número de níveis (L)	131
5.1.3	Fator de redução de malha (θ)	133
5.1.4	Fator de forte dependência na malha grossa (ε)	135
5.1.5	Número de incógnitas (N)	137
5.1.6	Otimização do algoritmo para o AMG	138
5.2	Equação de Laplace – Problema senoidal	139
5.2.1	Número de iterações internas (ν)	140
5.2.2	Número de níveis (L)	141
5.2.3	Fator de redução de malha (θ)	143
5.2.4	Fator de forte dependência na malha grossa (ε)	145
5.2.5	Número de incógnitas (N)	146
5.2.6	Otimização do algoritmo para o AMG	148
5.3	Comparativo para malhas triangulares	149
5.3.1	Número de iterações internas (ν)	150
5.3.2	Número de níveis (L)	150
5.3.3	Fator de redução de malha (θ)	151
5.3.4	Fator de forte dependência na malha grossa (ε)	152
5.3.5	Número de incógnitas (N)	152
5.3.6	Otimização do algoritmo para o AMG	153
5.4	Conclusões do Capítulo 5	154
6	Conclusão	156
	Referências	161
	Apêndice A – Definição do operador de interpolação	165
	Apêndice B – Estudo complementar para o fator de redução de malha	168

1 Introdução

Este capítulo descreve os problemas a serem abordados neste trabalho bem como sua importância em Dinâmica dos Fluidos Computacional (*Computational Fluid Dynamics - CFD*). Também é apresentada a motivação para a realização deste estudo, a revisão bibliográfica sobre o método *multigrid* algébrico (*Algebraic Multigrid Method - AMG*), os objetivos a serem alcançados e a organização do texto.

1.1 Generalidades em dinâmica dos fluidos computacional

A dinâmica dos fluidos envolve a modelagem de fenômenos físico-químicos que são representados por modelos matemáticos, sendo que estes fenômenos estão relacionados à mecânica dos fluidos, transferência de calor e massa, combustão, entre outros. A resolução destes problemas via métodos numéricos é o campo de estudos da dinâmica dos fluidos computacional.

Segundo Fortuna (2000), o objetivo em CFD é reduzir o número de experimentos e explorar fenômenos que não poderiam ser estudados em laboratório de forma prática. Para a solução de um problema em engenharia podem ser empregados três tipos de métodos: experimentais, analíticos e numéricos, sendo que cada um possui suas vantagens e desvantagens conforme o problema a ser resolvido. A Tab. 1.1, adaptada de Tannehill et al. (1997) compara os métodos acima citados.

De acordo com Ferziger e Peric (2002), o erro de modelagem é causado pelas simplificações feitas sobre o fenômeno real na concepção do modelo matemático. Os erros de modelagem afetam tanto as soluções analíticas quanto as numéricas, porque ambas se baseiam em modelos matemáticos.

Tabela 1.1: Comparação entre os métodos de solução de problemas em engenharia (Adaptada de Tannehill et al., 1997)

Método	Vantagens	Desvantagens
Experimental	mais realista	equipamento exigido, problemas de escala, dificuldades de medição, custo operacional
Analítico	mais geral, fórmula fechada	restrita a geometrias e processos físicos simples, geralmente restrita a problemas lineares, erros de modelagem
Numérico	não há restrição quanto à linearidade, geometrias e processos complicados, evolução temporal do processo	erros numéricos, prescrições das condições de contorno apropriadas, custo computacional, erros de modelagem

Para que seja possível tratar numericamente as equações diferenciais parciais, elas devem ser expressas na forma de operações aritméticas que o computador possa executar. Desta forma, recorre-se aos métodos numéricos para se obter uma solução aproximada, transformando o modelo contínuo em um modelo discreto.

Um método numérico pode resolver problemas mais gerais, mas em contrapartida pode apresentar diversas fontes de erros numéricos, que podem ser classificadas em (MARCHI, 2001):

- Erros de truncamento: Ocorrem quando o modelo matemático é aproximado por um método numérico. Quando se tem apenas erros de truncamento (as outras fontes de erro são desprezíveis), este erro denomina-se erro de discretização.
- Erros de iteração: É definido como sendo a diferença entre a solução exata das equações discretizadas e a solução numérica em uma determinada iteração (FERZIGER e PERIC, 2002).
- Erros de arredondamento: Ocorrem devido à representação finita dos números reais nos cálculos. Segundo Fortuna (2000), não podem ser evitados, mas podem ter seu efeito reduzido pela utilização de precisão dupla ou quádrupla nos cálculos.
- Erros de programação: São inerentes ao programador e à utilização do código computacional.

Para os problemas resolvidos neste trabalho, os erros de arredondamento são minimizados devido ao emprego de precisão dupla nos cálculos. Os erros de programação também são minimizados devido à verificação do código computacional através da análise de erros numéricos (mais detalhes no Capítulo 3). Desta forma, tem-se a presença de erros de iteração e de truncamento. O erro de iteração está ligado à aplicação do método *multigrid*, visto que os sistemas lineares resultantes da discretização das equações diferenciais são resolvidos com o emprego de métodos iterativos. Os erros de truncamento são provenientes da aproximação do modelo matemático por um método numérico.

O objetivo de um método numérico é obter a solução de uma ou mais equações diferenciais através da substituição das derivadas existentes por expressões algébricas que envolvam a função incógnita. O que caracteriza um método numérico é a forma como estas expressões são obtidas. Estas expressões são aplicadas a cada ponto do domínio de cálculo já discretizado, com as soluções obtidas nestes pontos. Existem várias formas desta discretização ser feita. As mais conhecidas são:

- Elementos Finitos (HUGHES, 2000; REDDY e GARTLING, 1994);
- Diferenças Finitas (FORTUNA, 2000; TANNEHILL et al., 1997);
- Volumes Finitos (MALISKA, 2004; VERSTEEG e MALALASEKERA, 2007).

Um método numérico, então, irá fornecer a solução em um número finito de pontos definidos pela malha computacional. Nestes casos, os problemas abordados dão origem a sistemas lineares do tipo:

$$Au = f \quad (1.1)$$

onde A é a matriz dos coeficientes, u é o vetor de incógnitas e f é o vetor independente.

Geralmente, quanto mais pontos nodais, mais próximo da solução analítica estará a solução numérica, pois o erro de truncamento (proveniente da discretização) será menor. De acordo com a distribuição destes pontos no domínio, a malha obtida pode ser uniforme (os pontos que compõem a malha estão uniformemente espaçados) ou não-uniforme; estruturada (uma estrutura ou regularidade na distribuição espacial dos pontos é apresentada) ou não-estruturada.

Quando o objetivo é reduzir os erros de truncamento, os problemas devem ser resolvidos em malhas extremamente refinadas. Para estes casos, o sistema linear é de grande porte. Em geral, a matriz de coeficientes é esparsa e, dependendo do tipo de malha, pode ser banda ou não (para malhas estruturadas ou não-estruturadas). No caso de sistemas de equações não-lineares, estes são representados por:

$$A(u) = f \quad (1.2)$$

Para resolver problemas de grande porte (consequência da discretização em malhas mais refinadas) com matrizes esparsas, são utilizados métodos iterativos, como por exemplo: Jacobi, Gauss-Seidel (BURDEN e FAIRES, 2009), ADI (*Alternate Direction Implicit Method*) (FERZIGER e PERIC, 2002), SOR (*Sucessive Over Relaxation*) (HACKBUSCH, 1985), MSI (*Modified Strongly Implicit Method*) (SCHNEIDER e ZEDAN, 1981), entre outros.

De acordo com Stüben (2001), a eficiência da solução numérica de grandes sistemas de equações requer uma hierarquia no algoritmo. Certamente um dos mais importantes avanços nas últimas três décadas é devido ao princípio do método *multigrid*. Este método tem duas abordagens, que estão relacionadas à forma de entrada dos dados e na construção das malhas auxiliares: o método *multigrid* geométrico (GMG) (Trottenberg et al, 2001 e Briggs et al, 2000), em que são necessárias informações a respeito das malhas auxiliares; e o método *multigrid* algébrico (AMG) (Ruge e Stüben, 1986; Brandt, 1986; Falgout, 2006 e Haase e Langer, 2002), em que é necessária a matriz de coeficientes. Desta matriz são retiradas informações a respeito da malha inicial, convencionada como a malha com mais incógnitas. O que os dois métodos têm em comum é o fato de o problema ser resolvido empregando uma hierarquia de malhas.

O método *multigrid* algébrico é composto por etapas que levam à resolução do problema, detalhadas a seguir. A resolução de um problema com o método *multigrid* algébrico, é dividida em fase *setup* e fase de solução (KRECHEL e STÜBEN, 1999, CLEARY et al., 2000, TROTTEBERG et al., 2001 e WATANABE et al., 2005). De acordo com Falgout (2006), os métodos *multigrid* alcançam a otimalidade empregando dois processos complementares: suavização e correção na malha grossa.

A fase *setup* é uma fase inicial, responsável pela geração das malhas, construção dos operadores de transferência entre as malhas (restrição e prolongação). A fase de solução é direta e emprega os operadores gerados na fase *setup* para se resolver o problema. Esta

resolução se dá com o uso de malhas auxiliares mais grossas (com menos pontos que a malha original), que são visitadas conforme um ciclo previamente determinado. O operador de restrição transfere as informações da malha fina para a malha imediatamente mais grossa, e o operador de prolongação transfere as informações da malha grossa para a malha imediatamente mais fina. Ciclo *multigrid* é a sequência com que as diversas malhas são visitadas, pode ser ciclo V, W, F, entre outros.

A suavização envolve a aplicação de um método iterativo (*solver*) em uma determinada malha, como por exemplo, Gauss-Seidel (também chamado método de relaxação ou suavizador). A correção na malha grossa envolve a resolução da equação residual nesta malha, obtendo uma aproximação para o erro. Esta aproximação é usada para corrigir a solução na malha imediatamente mais fina (o erro é transferido para esta malha). Fator de redução de malha é uma constante que quantifica se um ponto está fortemente conectado a outros. Fator de forte dependência na malha grossa é uma constante que define a forte dependência no conjunto dos pontos que estão na malha grossa. Estes dois últimos parâmetros interferem diretamente na construção das malhas auxiliares na aplicação do AMG.

1.2 O problema

Neste trabalho, os problemas abordados são descritos pelas equações de Laplace e Poisson bidimensionais, discretizadas em um domínio de cálculo quadrado, sendo que a solução analítica para estas equações é conhecida. Para a equação de Laplace são abordados dois problemas diferentes: um problema com condição de contorno dada por uma função linear a leste e ao norte, que neste trabalho é referenciado por problema linear. O outro problema tem como condição de contorno uma função senoidal ao norte, que neste trabalho é chamado de problema senoidal. Para todas as equações são consideradas condições de contorno de Dirichlet. A equação de Laplace é discretizada com o uso de malhas estruturadas quadrangulares e triangulares, que podem ser observadas nas Figs. 1.1 e 1.2, respectivamente. A equação de Poisson é discretizada somente com o uso de malhas estruturadas quadrangulares.

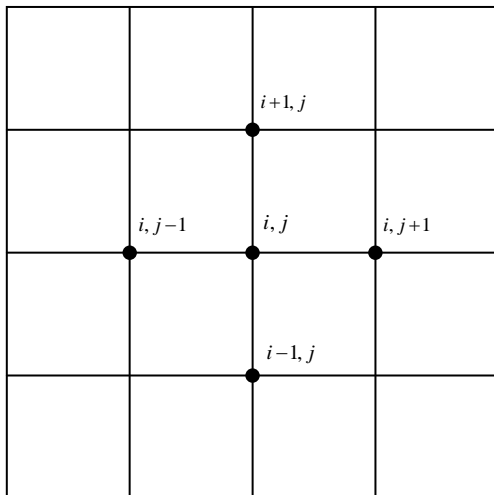


Figura 1.1: Malha quadrangular

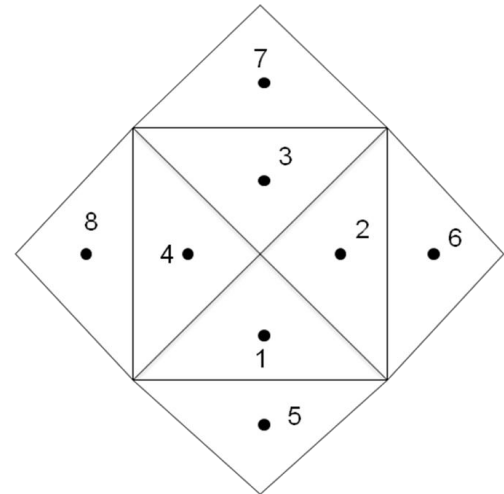


Figura 1.2: Malha triangular

Para a discretização das equações governantes em malhas quadrangulares, é empregado o método de diferenças finitas (MDF). Para malhas triangulares é utilizado o método dos volumes finitos (MVF), sendo que, para este caso, as condições de contorno são aplicadas com a utilização de volumes fictícios. Para a obtenção da solução numérica, é empregado o método *multigrid* algébrico (*Algebraic Multigrid* - AMG) a todos os problemas a serem resolvidos.

1.3 Motivação

Um dos campos mais atraentes de pesquisa em análise numérica é o desenvolvimento de métodos que operam hierarquicamente (com o uso de malhas auxiliares) para buscar, de forma eficiente, a solução de sistemas de equações com grande esparsidade, provenientes de malhas não-estruturadas (TROTTEBERG et al., 2001).

Muitas vezes a resolução de problemas de mecânica dos fluidos e transferência de calor, com o uso de métodos numéricos, torna-se inviável. Isto é devido ao grande número de equações a serem resolvidas a cada passo iterativo, requerendo assim, um alto custo computacional. Observa-se que, no início dos cálculos, a taxa de convergência é alta, passando a decair sensivelmente à medida que o processo iterativo avança (BRIGGS et al., 2000). Com o objetivo de melhorar a taxa de convergência dos métodos numéricos, vários trabalhos estão sendo desenvolvidos, sendo que um dos métodos utilizados com este objetivo é o método *multigrid* (BRANDT, 1986; BRIGGS et al., 2000, TROTTEBERG et al., 2001; STÜBEN, 2001; HAASE e LANGER, 2002; FALGOUT, 2006).

De acordo com Stüben (2001), este método opera sobre uma hierarquia de malhas, definida *a priori*, pelo engrossamento de uma dada discretização da malha, em um modo geometricamente natural (*multigrid* geométrico). Claramente, este procedimento é direto para malhas estruturadas. Porém, a definição de uma hierarquia de malhas pode se tornar complicada para malhas não-estruturadas.

A eficiência do método *multigrid* não tem sido totalmente alcançada em aplicações realísticas em CFD. Com a crescente complexidade destas aplicações, aumenta também a procura por métodos mais eficientes e robustos. É esperado que estes métodos tenham uma boa redução do tempo de CPU, usem pouca memória computacional e possam abordar as não-linearidades e acoplamentos, sem grandes prejuízos ao seu desempenho (RUGE e STÜBEN, 1986).

Um método que pode ser aplicado a problemas em que as localizações dos pontos são conhecidas, mas as malhas são não-estruturadas ou irregulares, é o método *multigrid* algébrico (BRANDT, 1986; RUGE e STÜBEN, 1986; STÜBEN, 2001; BRIGGS et al., 2000; TROTTEBERG et al., 2001 e FALGOUT, 2006). Este método também pode ser aplicado quando não se tem nenhuma informação a respeito da geometria do problema (STÜBEN, 2001). De acordo com Trottenberg et al. (2001), as vantagens deste método são robustez, aplicabilidade em situações com geometrias complexas e resolução de certos problemas que estão fora do alcance do *multigrid* geométrico.

Conforme Brandt (1986), o AMG pode ser usado como um *solver* “caixa-preta”. Como o algoritmo é baseado em uma dada matriz, ele pode ser usado para vários problemas. Nestes problemas, não há necessidade de se dar atenção especial para as condições de contorno, anisotropias, fortes descontinuidades em coeficientes, entre outros aspectos que normalmente requerem mais cuidado na discretização e na resolução do problema. Além disso, como a construção da matriz no AMG nos níveis auxiliares (nível com menos incógnitas) é baseada em fortes conexões algébricas, problemas que envolvem equações anisotrópicas, com fortes conectividades entre outros casos, podem também ser resolvidas pelo mesmo algoritmo AMG.

Outra aplicação do AMG é em problemas onde malhas não-estruturadas são empregadas e, por esse motivo, o método *multigrid* geométrico não é aplicável. Isto inclui discretização por elementos finitos com triangulações irregulares na malha com mais incógnitas, e matrizes de grandes dimensões que não são derivadas de problemas contínuos. Uma vantagem que o AMG apresenta sobre o *multigrid* geométrico é a forma como o

algoritmo faz a correção na malha grossa, engrossando apenas nas direções de suavização geométrica (FALGOUT, 2006).

De acordo com Trottenberg et al. (2001), o método *multigrid* (tanto geométrico quanto algébrico) é caracterizado por suas componentes: procedimentos de suavização (*solvers*), estratégia de engrossamento, operador de malha grossa, operadores de transferência de malhas finas para malhas grossas (e vice-versa) e tipo de ciclo.

Neste trabalho são abordados problemas com geometrias simples. Conforme Trottenberg et al. (2001), tais problemas não dão uma visão geral do método, mas permitem fácil investigação do comportamento assintótico do AMG, bem como sua dependência de vários aspectos específicos como anisotropias, descontinuidades, perturbações singulares, entre outros.

Conforme Trottenberg et al. (2001), uma simples modificação no algoritmo pode resultar em uma redução significativa no tempo de CPU. De acordo com Langer e Pusch (2006), para que o método *multigrid* seja realmente eficiente, é necessário adaptar as componentes do método de acordo com o delineamento físico do problema e a formulação variacional.

Com o desenvolvimento deste trabalho busca-se encontrar um algoritmo com parâmetros ótimos para a resolução de problemas com o uso do método *multigrid* algébrico. Entende-se por algoritmo com parâmetros ótimos como aquele que apresenta o menor tempo de CPU. Para este fim são observados parâmetros como: número de iterações internas (feitas pelo *solver*), número de níveis de malha a serem utilizadas na resolução do AMG, fator de redução de malha (constante que quantifica o quanto um ponto está fortemente conectado a outros) e fator de forte dependência na malha grossa.

Pretende-se também avaliar o efeito do número de incógnitas no tempo de CPU e fazer comparações entre os algoritmos considerados como padrão (parâmetros mais utilizados na literatura) e otimizado (com os parâmetros ótimos obtidos nos testes computacionais). Estas comparações envolvem o número de ciclos (iteraões externas) que o AMG leva para atingir a tolerância previamente determinada, e também o coeficiente c e o expoente p da equação $t = cN^p$, onde os valores para c e p serão determinados, N é o número de incógnitas e t é o tempo de CPU. Para esta última análise, (ajuste de curvas pelo método dos mínimos quadrados) são utilizados os tempos de CPU obtidos para os quatro maiores problemas resolvidos, de acordo com o tipo de malha empregado.

Neste trabalho é utilizado como base o código computacional AMG1R6 criado por Ruge e Stüben (1986). Esse programa resolve problemas discretizados por diferentes

métodos, tanto para malhas estruturadas quadrangulares quanto para malhas triangulares. O AMG1R6 traz implementado o *solver* Gauss-Seidel, o Gradiente Conjugado e o método direto YALE-SMP. O programa também pode ser executado para diferentes tipos de ciclos (V, F, W), com diferentes estimativas iniciais (nula, constante e igual a 1, randômica). O AMG pode ser usado também como um pré-condicionador. Neste código computacional são consideradas diversas modificações, para que os objetivos aqui propostos sejam alcançados.

1.4 Revisão bibliográfica

Desde o início dos anos 90, houve um forte aumento no interesse em métodos multiníveis algebricamente orientados. Uma razão para isto, certamente, foi o aumento da complexidade geométrica das aplicações que, tecnicamente, limitaram o uso do método *multigrid* geométrico (STÜBEN, 2001).

Conforme Chang et al. (1996), o método *multigrid* algébrico clássico foi introduzido por Brandt, McCormick e Ruge em 1982. Foi primeiro explorado por Stüben em 1983, e popularizado por Ruge e Stüben em 1986. De acordo com Ruge e Stüben (1986), o AMG pode ser usado para vários tipos de problemas, onde a aplicação do método *multigrid* geométrico é difícil ou impossível. Estes tipos de problemas estão listados a seguir:

- problemas onde a discretização na malha mais fina não permite engrossamento uniforme para todos os pontos;
- problemas causados pelo operador de transferência entre as malhas, e não pelo domínio. Quando o engrossamento uniforme é usado com interpolação linear, pode ser difícil encontrar um processo de relaxação que suavize o erro de forma suficiente para admitir uma boa correção na malha grossa. Algumas vezes, isto pode até não ser possível.
- problemas em que as informações são dadas apenas por uma matriz (não possuem informações geométricas).

O método *multigrid* algébrico resolve sistemas de equações baseado nos princípios do *multigrid* geométrico, mas de um modo que não requer conhecimento explícito da geometria do problema. O AMG determina as malhas grossas, os operadores de transferência entre as malhas e as equações de malha grossa baseado somente nas entradas da matriz.

De acordo com Falgout (2006), desde a introdução original do método, uma ampla variedade de algoritmos AMG tem sido desenvolvida. Estes algoritmos designam diferentes classes de problemas e possuem diferentes propriedades de robustez e eficiência.

Conforme Trottenberg et al. (2001), do ponto de vista espacial, o AMG tenta engrossar somente nas direções em que a relaxação realmente suaviza o erro. Porém, a informação relevante está contida na matriz (em termos da magnitude e dos sinais dos coeficientes). Este processo pode ser melhorado tendo como base apenas a informação da matriz, produzindo níveis grossos que são localmente adaptados para as propriedades de suavização.

Uma comparação entre os métodos *multigrid* geométrico e *multigrid* algébrico pode ser observada na Tab. 1.2 adaptada de Chang et al. (1996).

Tabela 1.2: Comparações entre *multigrid* geométrico e *multigrid* algébrico (Adaptado de CHANG et al., 1996)

	<i>Multigrid</i> Geométrico	<i>Multigrid</i> Algébrico
Problema a ser resolvido	Problemas Contínuos	Sistemas lineares de equações algébricas
Informação usada	Estrutura geométrica do problema	Somente as entradas da matriz
Programa	Preparado para cada problema	Único para todos os problemas
Eficiência	Muito boa	Boa

A equação de Laplace foi resolvida em Ruge e Stüben (1986), que apresentaram como resultados o tempo da fase *setup* e de um ciclo V. Demonstraram a robustez e a eficiência do método *multigrid* algébrico em uma variedade de casos, incluindo domínios irregulares.

Dietrich (1995) resolveu a equação de Poisson em malhas não-estruturadas e apresentou como resultados o número de iterações (ciclos) e o tempo de CPU (fase *setup* junto com ciclo V). Para a obtenção dos resultados, considerou 4 níveis de malha (fez 3 engrossamentos), sendo que o sistema de equações na malha mais grossa tem de 30 a 50 incógnitas. Considerou como otimização o uso do método *multigrid* como pré-condicionador e seus resultados foram comparados aos obtidos com o método de Cholesky incompleto usado como pré-condicionador. O AMG obteve os melhores resultados quando comparados o número de iterações e o tempo de CPU.

Outros autores também apresentaram estudos para a equação de Poisson com AMG. Chang et al. (1996) trabalharam com malhas estruturadas e apresentaram resultados para o tempo de um ciclo V e também para a fase *setup*. Utilizaram uma nova formulação para

interpolação, que não somente melhorou a taxa de convergência, como também pôde ser aplicada a casos mais gerais, incluindo matrizes que não sejam diagonal dominante.

Krechel e Stüben (1999) aplicaram AMG à equação de Poisson, onde o principal objetivo foi o de analisar o fator de convergência de um ciclo V. Testaram duas formas de interpolação combinadas com diferentes formas de se aplicar os passos de suavização.

Briggs et al. (2000) aplicaram o AMG à equação de Poisson com o uso de malhas estruturadas e compararam os resultados aos obtidos com o método *multigrid* geométrico. Apresentaram a norma obtida em cada ciclo V conforme o avanço do processo iterativo. Concluíram que, em termos de performance, o AMG é equivalente ao *multigrid* geométrico.

Cleary et al. (2000) mostraram os resultados obtidos para a equação de Poisson, tanto para malhas estruturadas quanto para malhas não-estruturadas. Em seus resultados, apresentaram os tempos obtidos para um ciclo V e também para a fase *setup*. Para malhas estruturadas, o fator de convergência não mostrou dependência do tamanho da malha, e os tempos obtidos para a fase *setup* e para o ciclo V apresentaram crescimento linear com o número de incógnitas. Para malhas não-estruturadas (obtidas por triangulação uniforme), o fator de convergência mostrou dependência com o tamanho da malha.

Trottenberg et al. (2001) resolveram problemas tipo Poisson com o uso do AMG em malhas estruturadas. Apresentaram resultados para tempos de CPU e número de ciclos até a tolerância estipulada em 10^{-10} ser atingida. Apresentaram também resultados para o fator de convergência para diferentes métodos. De acordo com os autores, é recomendada a aplicação do AMG a equações simples (Equações de Poisson ou de Laplace) para o estudo de parâmetros que compõem o método. No presente trabalho, os resultados obtidos para a equação de Laplace são comparados aos obtidos para a equação de Poisson em malhas quadrangulares estruturadas.

O trabalho de Iwamura et al. (2003) aplicou o AMG1R6 de Ruge e Stüben (1986) a equações que foram discretizadas com o método dos elementos finitos, em malhas não-estruturadas em três dimensões. O problema resolvido neste trabalho envolve a simulação numérica de escoamentos (modelagem da injeção de polímeros). Foram apresentadas comparações entre diversos algoritmos utilizados, entre eles AMG1R6 padrão e AMG1R6 com gradiente conjugado pré-condicionado. Empregaram diferentes valores para o fator de redução de malha, dependendo do nível de malha que estava sendo tratado. Empregaram também diferentes valores para o fator de forte dependência na malha grossa, dependendo de quantos níveis estavam sendo utilizados para a resolução do problema. Concluem que sistemas simples nas malhas grossas (com operadores de complexidade pequenos) são obtidos

reduzindo substancialmente os sistemas no nível grosso pelo uso de um conjunto pequeno de pontos na malha grossa.

O trabalho de Pereira et al. (2006) apresentou uma nova abordagem para selecionar as malhas grossas. Esta abordagem foi baseada em uma escolha apropriada para o fator de redução de malha, que considerou a densidade da matriz durante o processo de engrossamento. Este procedimento levou a uma redução da dimensão da matriz para todos os níveis, sem perder a robustez do método. Empregaram o AMG pré-condicionado com o método gradiente conjugado.

Falgout (2006) resolveu a equação de advecção-difusão, discretizada com o método dos elementos finitos em malhas uniformes. Como resultados apresentou o número de iterações, níveis de malha, tempo para a fase *setup* e tempo para resolver o problema. Os resultados obtidos neste trabalho foram os esperados para o método *multigrid*. O fator de convergência independe do tamanho do problema e o aumento do tempo para resolver o problema e do tempo para a fase *setup* apresentaram crescimento linear com o número incógnitas.

O trabalho de Oliveira et al. (2008) apresentou resultados para a equação de Laplace, discretizada em malhas quadrangulares, resolvida com o método *multigrid* geométrico. Apresentou um estudo sobre o número de iterações internas (feitas pelo *solver*) e sobre o número de níveis, encontrando valores ótimos para estes parâmetros. Pretende-se comparar qualitativamente esses resultados aos obtidos no presente trabalho com o método *multigrid* algébrico.

O trabalho de Gaspar et al. (2009) trouxe resultados para o método *multigrid* geométrico aplicado a malhas triangulares, discretizadas com o método dos elementos finitos. Compararam duas estratégias de engrossamento. Fizeram testes para diferentes tipos de ciclos (V, W e F) e também variaram o número de iterações feitas pelo *solver*. Compararam número de iterações externas e número de operações aritméticas. Pelo número de operações aritméticas concluíram que o melhor algoritmo para o método *multigrid* geométrico é o ciclo V com uma iteração interna para o *solver*.

Em todos os trabalhos anteriormente citados, não se encontrou um estudo para a obtenção do número ótimo de iterações internas para o método *multigrid* algébrico. Também não se tem dados sobre o número ótimo de malhas (níveis) a ser utilizado no AMG, apesar do valor para este parâmetro estar disponível em alguns trabalhos. Dietrich (1995) utilizou malhas com 4 níveis, Krechel e Stüben (1999), utilizaram 7 níveis, e resolveram até a malha mais grossa considerada, com 9 variáveis. Trottenberg et al. (2001) apresentaram estudos para

dois níveis de malha, Dumett et al. (2002) resolveram problemas com 7, 8, 9 e 11 níveis, Reitzinger et al. (2003), utilizaram 6 níveis mas não foram até a malha mais grossa possível e Falgout (2006) resolveu problemas com 5, 6, 8, 9, 12 e 13 níveis. Briggs et al. (2000), Cleary et al. (2000) e Pereira et al. (2006) utilizaram vários tamanhos de problemas em seus testes computacionais, sendo que para todos eles, foram utilizados todos os níveis de malha.

Na literatura também é observada discordância com relação aos critérios de parada e tolerâncias utilizadas para o AMG. Dietrich et al. (1995) empregaram o resíduo baseado na norma l_2 e levaram o processo iterativo até que se atinja a tolerância de 10^{-5} . Chang et al. (1996) calcularam o resíduo adimensionalizado pela estimativa inicial e mantiveram a tolerância fixa em 10^{-6} , assim como Falgout (2006) que empregou a tolerância de 10^{-9} . Cleary et al. (2000) calcularam a norma do resíduo de um ciclo para outro. Nenhum desses autores informa como foi determinada a tolerância adotada. Neste trabalho, o critério de parada baseado na norma l_2 do resíduo, adimensionalizada pela norma na estimativa inicial, é o adotado. Este critério de parada, além de ser empregado por Briggs et al. (2000) e Trottenberg et al. (2001), ainda é o utilizado por Oliveira et al. (2008), que terão seus resultados comparados aos obtidos nesta tese. A tolerância é determinada para cada um dos problemas abordados. Mais detalhes sobre a obtenção da tolerância para cada um dos problemas são apresentados no Capítulo 3.

Observa-se que não se tem estudos para a influência do fator de redução da malha (θ) sobre o tempo de CPU. Este número, que é uma constante entre 0 e 1, ($0 < \theta < 1$), quantifica se um ponto está fortemente conectado a outros. Este parâmetro influencia diretamente a geração das malhas auxiliares para a aplicação do AMG. Usualmente é adotado $\theta = 0,25$, conforme é recomendado em Trottenberg et al. (2001), que afirmaram que este é um valor padrão razoável. Ruge e Stüben (1986), Chang et al. (1996), Krechel e Stüben (1999) e Cleary et al. (2000) também utilizaram $\theta = 0,25$. Por analogia ao GMG, considerando uma malha estruturada, $\theta = 0,25$ ou $\theta = 1/4$ corresponde à interpolação de quatro pontos (norte, sul, leste e oeste) para se obter o ponto da malha imediatamente mais fina, o que corresponde à razão de engrossamento $r = 2$ para o GMG. Briggs et al. (2000) utilizaram $\theta = 0,2$ e Falgout (2006) empregou $\theta = 0,4$.

Pereira et al. (2006) empregaram vários valores de θ , pois seus dados de entrada estão na forma matricial (estes valores foram obtidos por meio de um cálculo que leva em conta a densidade da matriz). Os resultados de Pereira et al. (2006) sugerem que o uso de pequenos

valores para θ produzem uma grande redução na malha e no operador de complexidade. Em consequência a isso, há uma redução nos tempos para a fase *setup* e para a fase de solução do AMG. Porém o uso de pequenos valores para θ pode aumentar o fator de convergência, reduzindo a robustez do método.

Para Campos et al. (2006), a escolha de θ influencia diretamente o número de pontos da malha em cada nível, ou seja, o número de entradas não-nulas de cada matriz. Grandes valores para θ geram matrizes nas malhas grossas com mais informações a respeito das malhas mais finas. Isto contribui para a rápida convergência em termos do número de iterações. Por outro lado, esta melhora não resulta em menores tempos de CPU, já que cada nível é computacionalmente mais custoso.

Não foram encontrados estudos da influência do fator de forte dependência na malha grossa (ε) no tempo de CPU. Este parâmetro é uma constante e seu valor, que deve ser positivo, está fixado em $\varepsilon = 0,35$, conforme encontrado em Ruge e Stüben (1986), Chang et al. (1996), Krechel e Stüben (1999), Briggs et al. (2000), Cleary et al. (2000) e Falgout (2006). Em Trottenberg et al. (2001) foi utilizado $\varepsilon = 0,20$ e em Iwamura et al. (2003), $\varepsilon = 0,35$ e $\varepsilon = 0,45$, de acordo com o nível de malha que estava sendo resolvido.

Alguns dos parâmetros aqui estudados são comparados com os ótimos obtidos para o método *multigrid* geométrico, encontrados em Oliveira et al. (2008). Existem alguns trabalhos disponíveis na literatura, que trazem comparações entre AMG e GMG. Resultados numéricos podem ser encontrados em Briggs et al. (2000), que apresentaram comparações entre o método *multigrid* algébrico e geométrico para a equação de Poisson 2D, discretizada em malhas estruturadas quadrangulares, com o método de diferenças finitas. Deste exemplo, concluiu-se que a norma obtida em cada ciclo V com o AMG tem o mesmo fator de redução que o encontrado para o *multigrid* geométrico. Concluiu-se ainda que, para o caso de malhas estruturadas, o AMG utiliza mais memória computacional e gasta mais tempo de CPU na resolução do problema, quando comparado ao *multigrid* geométrico (a fase *setup* do AMG gasta sete vezes mais tempo que um ciclo V (tempo total) do GMG, para a malha com 64 incógnitas).

Haase e Langer (2002) compararam os resultados obtidos para o método *multigrid* algébrico com os obtidos para o *multigrid* geométrico, para um problema de campo magnético. Usaram apenas 3 malhas, tanto para o AMG quanto para o GMG, sendo que a malha considerada grossa para o AMG teve menos de 500 incógnitas, e para o GMG teve

2907 incógnitas. Analisaram o tempo de CPU, número de iterações e *speedup*, para o problema resolvido com computação paralela.

O trabalho de Watanabe et al. (2005) comparou os tempos de CPU para os dois métodos. Concluíram que os tempos obtidos para o AMG e o GMG aumentam linearmente com o número de incógnitas, sendo que o GMG é aproximadamente 3 vezes mais rápido que o AMG.

Langer e Pusch (2006) compararam os dois métodos (AMG e GMG), pré-condicionados, para a equação de Laplace em 3 dimensões. Fizeram comparações do número de ciclos para o AMG e o GMG.

Wu e Elman (2006) resolveram a equação de advecção-difusão discretizada com o método de elementos finitos, empregando AMG e GMG, em malhas adaptativas e uniformes. Compararam o número de iterações que cada método levou para convergir. Concluíram que a taxa de convergência deteriora quando os parâmetros difusivos diminuem.

Campos et al. (2006) compararam a performance do AMG com GMG, ambos pré-condicionados e com algoritmos paralelos para um sistema não-linear de equações diferenciais. Apresentaram resultados para número de níveis do GMG e número de níveis e fator de redução de malha para o AMG, quando são utilizados diferentes números de processadores. O número de níveis do GMG variou de 2 até 6, e do AMG, de 6 até 16. Os valores testados para o fator de redução de malha foram: 0,25; 0,50 e 0,75. Os resultados mostraram que, para o GMG, o número de níveis ótimo, para um processador, é 2. Para o AMG, o melhor fator de redução de malha é 0,25, e o número de níveis ótimo é 8.

1.5 Objetivos

Este trabalho tem por objetivo otimizar os parâmetros do algoritmo do método *multigrid* algébrico, através da avaliação de seus parâmetros em malhas estruturadas quadrangulares e triangulares. Comparações qualitativas são feitas com os parâmetros já conhecidos para o método *multigrid* geométrico. O método *multigrid* algébrico é aplicado às equações de Laplace e Poisson. Os objetivos gerais do trabalho são:

- Analisar os parâmetros do método *multigrid* algébrico em malhas estruturadas quadrangulares e triangulares.

- Minimizar o tempo de CPU para o método *multigrid* algébrico em malhas estruturadas quadrangulares e triangulares para as equações de Laplace e Poisson.

Os objetivos específicos são:

Para a equação de Laplace, em um domínio de cálculo quadrado, discretizado com malhas estruturadas quadrangulares e triangulares e para a equação de Poisson, discretizada em um domínio de cálculo quadrado com malhas estruturadas quadrangulares, pretende-se:

- Encontrar o número ótimo de iterações internas;
- Encontrar o número ótimo de malhas (níveis);
- Encontrar o valor ótimo para o fator de redução de malha;
- Encontrar o valor ótimo para o fator de forte dependência na malha grossa;
- Avaliar o efeito do número de incógnitas no tempo de CPU;
- Avaliar os parâmetros acima quando alteradas as condições de contorno para a equação de Laplace;
- Comparar os parâmetros ótimos encontrados para malhas estruturadas quadrangulares com os obtidos para malhas estruturadas triangulares;
- Comparar os parâmetros ótimos encontrados para o método *multigrid* algébrico em malhas estruturadas quadrangulares com os já conhecidos para o método *multigrid* geométrico.
- Comparar os parâmetros ótimos obtidos para a equação de Laplace com os obtidos para a equação de Poisson, ambas com o AMG em malhas estruturadas quadrangulares;
- Obter resultados em malhas triangulares mais refinadas que as apresentadas na literatura.

1.6 Organização do texto

Neste capítulo foram descritas as generalidades em dinâmica dos fluidos computacionais, além de ter apresentado os problemas a serem abordados nesta tese. Foi

apresentada ainda a motivação para a realização deste trabalho, além da revisão bibliográfica, que citou os trabalhos mais relevantes para a definição dos objetivos desta tese.

No próximo capítulo é apresentada a fundamentação teórica sobre os métodos iterativos para a solução de sistemas de equações lineares, sobre o critério de convergência a ser adotado, e a teoria do método *multigrid* algébrico. Tem-se também as formulações para o método dos volumes finitos aplicado a malhas estruturadas triangulares e para o método de diferenças finitas aplicado à malhas quadrangulares. No capítulo 3 é apresentada uma análise dos códigos computacionais empregados na resolução das equações diferenciais em questão. Esta análise é feita através do estudo de erros numéricos. No capítulo 4 são apresentados os resultados obtidos em malhas quadrangulares, para as equações de Laplace e Poisson. No capítulo 5 têm-se os resultados obtidos em malhas triangulares, para a equação de Laplace. No capítulo 6 são apresentadas as conclusões, as contribuições desta tese, algumas recomendações práticas para o uso do AMG, além de sugestões para trabalhos futuros. No apêndice A tem-se uma demonstração do operador de interpolação para o AMG, e no apêndice B é apresentado um estudo complementar para o fator de redução de malha para o problema linear em malhas quadrangulares.

2 Fundamentação teórica

Neste capítulo é apresentada uma revisão dos métodos iterativos básicos e a fundamentação teórica do método *multigrid* algébrico. Na sequência, têm-se as formulações utilizadas na discretização das equações diferenciais, tanto com o uso do método de diferenças finitas, para malhas estruturadas quadrangulares, quanto para o método dos volumes finitos, para malhas estruturadas triangulares. São apresentados também os modelos matemáticos e numéricos aqui resolvidos, bem como as variáveis de interesse consideradas para a verificação do código computacional, utilizadas no Capítulo 3.

2.1 Métodos iterativos e análise de convergência

Os métodos numéricos utilizados para a obtenção da solução de sistemas lineares podem ser divididos em dois grupos (CUNHA, 2003): métodos diretos (que conduzem à solução exata, a menos de erros de arredondamento introduzidos pela máquina após um número finito de passos) e métodos iterativos (que são baseados na construção de seqüências de aproximações). Nos métodos iterativos, a cada passo, os valores calculados anteriormente são usados para melhorar a aproximação.

Os métodos diretos podem ser inviáveis quando o sistema de equações é muito grande ou mal condicionado. Por outro lado apresentam a vantagem de fornecer a solução após um número finito de passos e de não dependerem de critérios de convergência. Os métodos diretos mais utilizados são (FORTUNA, 2000): eliminação de Gauss, fatoração LU, TDMA (*TriDiagonal Matrix Algorithm*), sendo este último aplicado a sistemas tridiagonais de equações.

Os métodos iterativos geram uma seqüência de vetores a partir de uma aproximação inicial. Sob certas condições esta seqüência converge para a solução exata, caso ela exista. Segundo Fortuna (2000), os métodos iterativos apresentam a vantagem de serem aplicáveis

quando o sistema de equações é não-linear. De acordo com Burden e Faires (2009), estes métodos apresentam vantagens quando se tem problemas com muitas incógnitas, além de aproveitarem a esparsidade da matriz de coeficientes. Alguns exemplos de métodos iterativos: método de Jacobi, Gauss-Seidel, SOR, ADI, MSI. Neste trabalho, o método empregado para a resolução de sistemas de equações algébricas (*solver*), é o método de Gauss-Seidel com ordenamento lexicográfico. Este método é o utilizado, pois é o mais empregado pela literatura além de ser de fácil implementação. A aplicação deste método a uma equação qualquer está descrita a seguir.

Dada uma equação diferencial governante para o caso bidimensional, uma equação algébrica é formada para cada volume de controle considerando o método dos volumes finitos (Eq. 2.1). Veja abaixo o exemplo de um esquema de 5 pontos, onde a localização dos pontos pode ser observada na Fig. 2.1:

$$a_{i,j}u_{i,j} = a_{i,j}^n u_{i+1,j} + a_{i,j}^s u_{i-1,j} + a_{i,j}^e u_{i,j+1} + a_{i,j}^w u_{i,j-1} + f_{i,j} \quad (2.1)$$

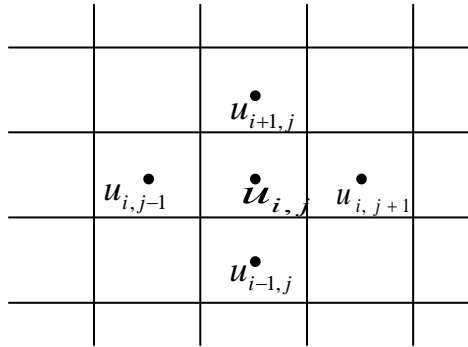


Figura 2.1: Malha geométrica utilizada para Gauss-Seidel

Na Eq. (2.1), $a_{i,j}$ são os coeficientes, $u_{i,j}$ são as variáveis nodais ou nós do volume de controle e $f_{i,j}$ é o termo fonte. O método de Gauss-Seidel é um método iterativo usado para resolver este tipo de sistema de equações. Este método resolve o sistema visitando equação por equação, iterativamente, usando em um mesmo ciclo os valores das variáveis já calculadas neste ciclo iterativo. Pode-se reescrever a Eq. (2.1) na forma:

$$a_{i,j}u_{i,j}^{m+1} = a_{i,j}^n u_{i+1,j}^m + a_{i,j}^s u_{i-1,j}^{m+1} + a_{i,j}^e u_{i,j+1}^m + a_{i,j}^w u_{i,j-1}^{m+1} + f_{i,j} \quad (2.2)$$

onde o superíndice m representa a m -ésima iteração e os subíndices i, j representam a posição do nó na malha computacional. Neste caso, como foi utilizado ordenamento lexicográfico, tem-se as incógnitas tomadas de oeste para leste e do sul para o norte, podendo-se considerar como conhecidas, na mesma iteração, as variáveis $u_{i,j-1}$ e $u_{i-1,j}$. O procedimento para o método Gauss-Seidel está descrito na Tab. 2.1.

Tabela 2.1: Procedimento iterativo Gauss-Seidel (BURDEN e FAIRES, 2009)

Procedimento Gauss-Seidel para $Au = f$

- 1) Dados: $m=0$, m_maximo e u^0 ;
- 2) Calcular $u_{i,j}^{m+1}$ pela Eq. (2.2) para todo i, j ;
- 3) Faça m receber $m+1$;
- 4) Volte ao passo 2 até convergir ou atingir m_maximo .

Segundo Ferziger e Peric (2002), quando o método utilizado para se resolver os sistemas de equações é iterativo, como o adotado neste trabalho, é importante adotar um critério de parada, conforme descrito a seguir.

Basicamente, deve-se lembrar que a discretização de uma equação diferencial através de um método numérico resulta na Eq. (1.1), que será repetida aqui para fins didáticos:

$$Au = f \quad (2.3)$$

O resíduo para o sistema apresentado pela Eq. (2.3) é definido genericamente por (Ferziger e Peric, 2002)

$$r^m = f - Au^m \quad (2.4)$$

onde u^m é a solução da incógnita na iteração m e r^m é o resíduo na iteração m . Para o caso particular que será tratado neste trabalho, tem-se:

$$r^m = a_{i,j-1}u_{i,j-1}^m + a_{i,j+1}u_{i,j+1}^m + a_{i+1,j}u_{i+1,j}^m + a_{i-1,j}u_{i-1,j}^m - a_{i,j}u_{i,j}^m \quad (2.5)$$

O critério para interromper o processo iterativo com base no resíduo das equações é dado por:

$$\frac{L^m}{L^0} \leq Tol \rightarrow \text{parar} \quad (2.6)$$

onde L^m é a norma dos resíduos na iteração m ; L^0 é a norma dos resíduos na estimativa inicial. Neste caso, os resíduos calculados são adimensionalizados pela norma da estimativa inicial, e Tol , é a tolerância admitida para interromper o processo iterativo. A norma empregada pode ser norma l_1 , norma Euclidiana (ou norma l_2), norma do máximo (ou norma l_∞) (KREYSZIG, 1999) entre outros. Neste trabalho é empregada a norma l_2 do resíduo, dada pela Eq. (2.7):

$$L^m = \|r^m\|_2 = \sqrt{\sum_{j=1}^N (r_j^m)^2} \quad (2.7)$$

onde N é o número de incógnitas e m é a iteração. Por questões práticas, o critério de parada adotado neste trabalho é referenciado por norma l_2 do resíduo nas citações do texto que seguem. A tolerância é estimada no próximo capítulo, sendo que seu valor é obtido com base em testes computacionais, que tem o processo iterativo levado até o erro de máquina ser atingido.

2.2 Equação residual

No contexto do método *multigrid*, a equação residual tem relevante importância para a obtenção da solução do sistema de equações. Em sistemas lineares, para a obtenção da equação residual, deve-se supor que o sistema dado pela Eq. (2.3) possui solução única, e que v é uma aproximação calculada para u . Conforme Briggs et al. (2000), existem duas importantes medidas de v como uma aproximação de u . Uma delas é o erro (também chamado de erro algébrico) dado por:

$$e = u - v \quad (2.8)$$

A magnitude do vetor do erro pode ser medido através de normas de vetores (BURDEN e FAIRES, 2009). As normas mais utilizadas são: norma do máximo (ou norma

infinito) e norma Euclidiana, ambas do erro, definidas pelas Eqs. (2.9) e (2.10), respectivamente:

$$\|e\|_{\infty} = \max_{1 \leq j \leq N} |e_j| \quad (2.9)$$

$$\|e\|_2 = \sqrt{\sum_{j=1}^N e_j^2} \quad (2.10)$$

Muitas vezes, este erro é tão inacessível quanto a solução exata do sistema. Porém, uma medida calculável de como v se aproxima de u , é o resíduo dado por:

$$r = f - Av \quad (2.11)$$

O resíduo é uma medida de quanto a aproximação v falha ao satisfazer o problema original. Seu módulo também pode ser medido através de normas de vetores. A relação entre erro e resíduo é dada pela equação residual:

$$Ae = r \quad (2.12)$$

Esta equação informa que o erro satisfaz o mesmo conjunto de equações com a variável u , quando f é substituída pelo resíduo r . A equação residual apresenta uma grande vantagem. Supondo que uma aproximação v seja determinada por algum método iterativo, o resíduo é calculado através da Eq. (2.11). Para se obter uma melhor aproximação v , a equação residual (Eq. (2.12)) é resolvida para e e então uma nova aproximação é calculada usando a definição do erro:

$$u = v + e \quad (2.13)$$

Para sistemas de equações não-lineares, dados pela Eq. (1.2), tem-se que o resíduo é dado por:

$$r = f - A(v) \quad (2.14)$$

E a relação entre o erro ($e = u - v$) e o resíduo é dada pela equação residual:

$$A(u) - A(v) = r \quad (2.15)$$

A Eq. (2.15) recai na Eq. (2.12) quando se tem sistemas lineares.

2.3 Método *multigrid* algébrico

Nesta seção é apresentada a teoria para o método *multigrid* algébrico (AMG). São expostos os princípios fundamentais deste método e os procedimentos aplicados para se fazer o engrossamento e a prolongação. São apresentados também os algoritmos empregados na resolução de problemas com o uso do AMG.

2.3.1 Princípios fundamentais

O método *multigrid* é composto por dois princípios básicos (WESSELING, 1992 e BRIGGS et al., 2000): princípio de suavização e princípio de malha grossa. O princípio de suavização assegura que métodos iterativos clássicos (como Gauss-Seidel, por exemplo), quando aplicados apropriadamente a problemas elípticos discretos, têm um forte efeito de suavização sobre o erro de qualquer aproximação. O princípio de malha grossa afirma que, um termo de erro suave é bem aproximado em uma malha grossa. Um procedimento em malha grossa gasta menos tempo de CPU e usa menos memória computacional que um procedimento em malha fina.

Grande parte dos métodos iterativos padrão apresenta propriedades de suavização de erros locais de alta frequência (componentes oscilatórios do erro), enquanto as baixas frequências são mantidas praticamente inalteradas. Deste modo, as primeiras iterações do processo, geralmente apresentam rápida convergência, caracterizando a presença de modos oscilatórios do erro. Porém, após algumas iterações a convergência torna-se lenta, sinalizando a presença de modos suaves (BRANDT, 1977; WESSELING, 1992).

Para ilustrar este procedimento, é resolvido o exemplo encontrado em Briggs et al. (2000). A equação de Laplace unidimensional, discretizada com o método de diferenças finitas, é resolvida com o uso do método de Jacobi ponderado (com $\omega = 2/3$). Para a

resolução deste problema, a estimativa inicial aplicada é dada pelos modos de Fourier, que são dados pela Eq. (2.16). Os modos de Fourier para alguns valores de k podem ser observados na Fig. 2.2.

$$v_j = \text{sen}\left(\frac{jk\pi}{n}\right), \quad 0 \leq j \leq n, \quad 1 \leq k \leq n-1 \quad (2.16)$$

onde j é a componente do vetor \mathbf{v} , n é o número de elementos e k é o número de ondas ou modos de Fourier. Na Fig. 2.2 podem ser observados os modos de Fourier para $k=1$, $k=2$, $k=6$. Os valores para k denotam quantos “meio-senos” constituem a curva \mathbf{v} no domínio do problema. A notação \mathbf{v}_k indica um vetor \mathbf{v} que contém k ondas.

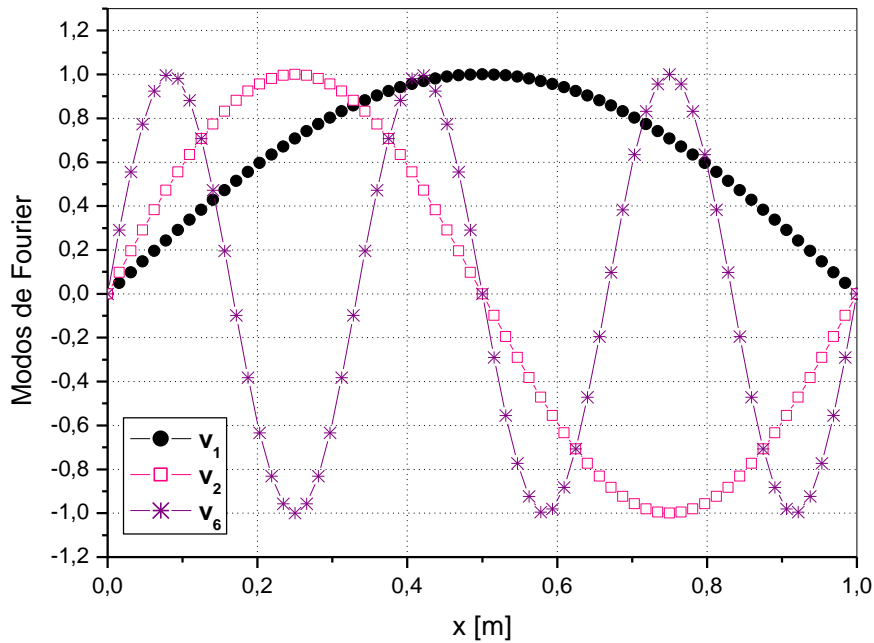


Figura 2.2: Modos de Fourier

Nota-se na Fig. 2.2 que pequenos valores de k correspondem a ondas longas e suaves, enquanto que valores maiores de k se referem a ondas mais curtas e oscilatórias. De acordo com Briggs et al. (2000) e Trottenberg et al. (2001), os métodos iterativos, como Jacobi e Gauss-Seidel, possuem propriedades de suavização, ou seja, são capazes de reduzir rapidamente as componentes de erros oscilatórios (ou de alta frequência).

Os modos de Fourier localizados na metade inferior do espectro, com $1 \leq k < N/2$, são chamados de modos de Fourier de baixa frequência ou modos suaves. Os modos de Fourier localizados na metade superior do espectro, $N/2 \leq k < N-1$, são chamados de modos de

Fourier de alta frequência ou modos oscilatórios. Nota-se então que a definição dos modos de Fourier como sendo modos suaves ou oscilatórios, depende do número de incógnitas do problema que está sendo resolvido.

A equação de Laplace, discretizada com o método de diferenças finitas, é resolvida agora com estimativa inicial dada pela combinação de dois modos de Fourier ($\mathbf{v}_2 + \mathbf{v}_{16}$). O domínio de cálculo é discretizado com dois tamanhos de malhas diferentes: uma malha com 128 incógnitas e a outra com 32 incógnitas. Na Fig. 2.3 (a) pode-se observar a solução para esta equação na malha mais refinada, enquanto que na Fig. 2.3 (b), está ilustrada a solução obtida para a malha mais grossa. Observa-se que, para as duas malhas, são consideradas três situações: a estimativa inicial, o resultado após 10 e após 100 iterações.

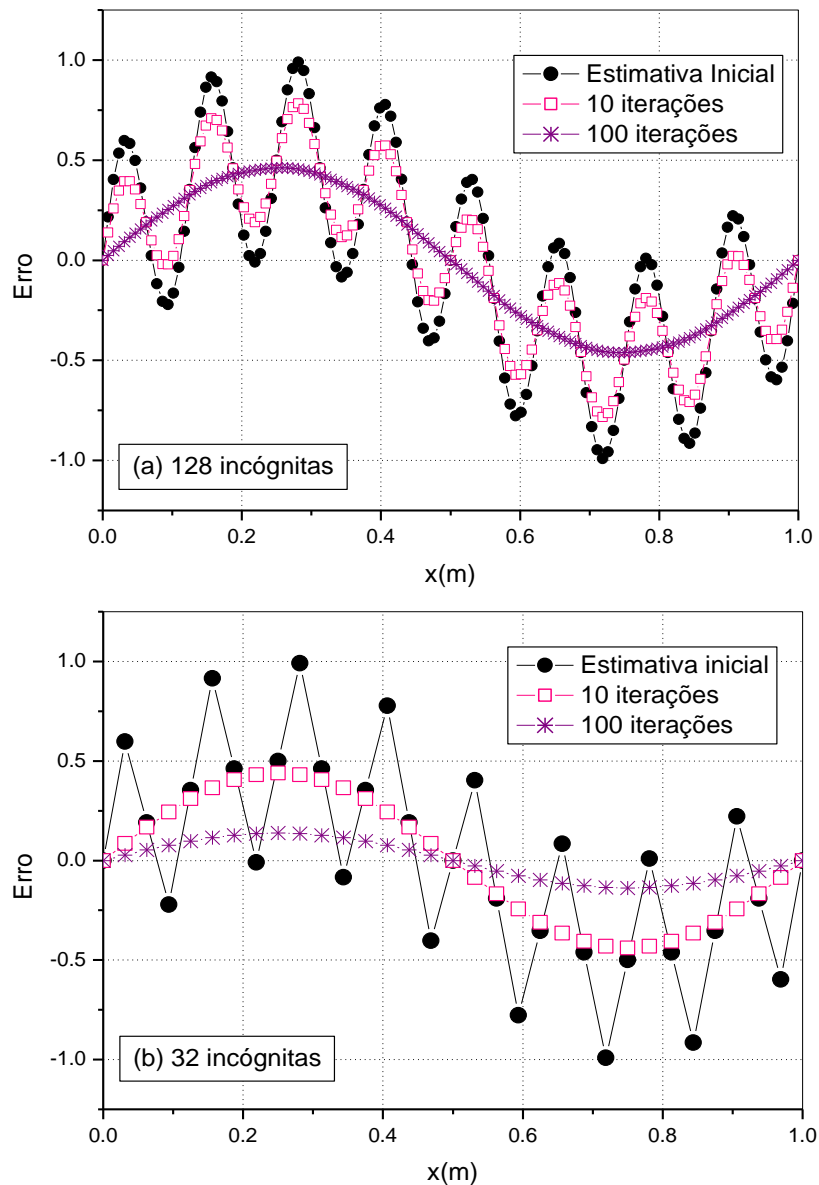


Figura 2.3: Comportamento da suavização do erro em métodos iterativos

Na malha mais refinada, Fig. 2.3 (a), após 10 iterações, nota-se a suavização das componentes de alta frequência, enquanto que as de baixa frequência permanecem praticamente inalteradas. Após 100 iterações, nesta malha, nota-se que os erros que eram de alta frequência estão suavizados, enquanto que, as componentes de baixa frequência, são pouco afetadas por esta propriedade de suavização. Já na Fig. 2.3 (b), para a malha mais grosseira, depois de 10 iterações, nota-se a suavização das componentes de alta e baixa frequência. Após 100 iterações, esta suavização fica ainda mais evidente, tanto para as componentes de alta quanto de baixa frequência.

Na Fig. 2.4, obtida de Briggs et al. (2000) pode-se observar como as componentes de erro suave apresentam-se em uma malha fina e em uma malha grossa. Na parte superior da Fig. (2.4) tem-se o modo de Fourier para a malha fina e na inferior, o mesmo modo de Fourier projetado na malha imediatamente mais grossa.

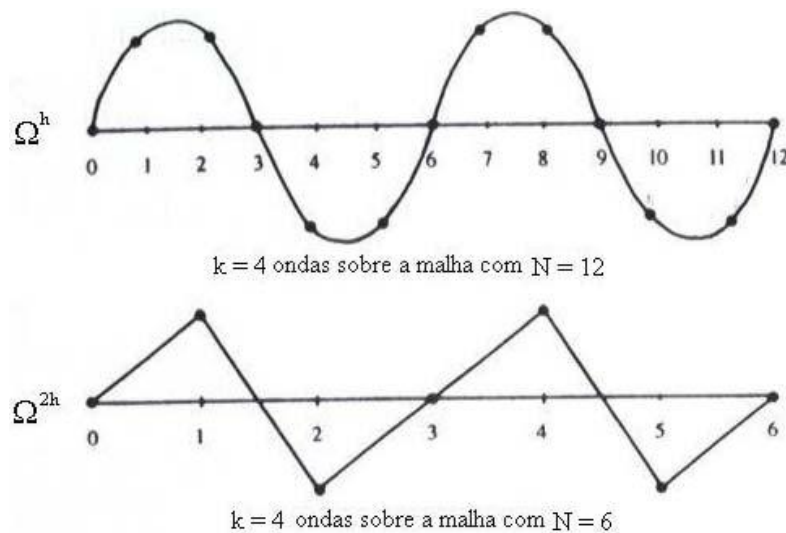


Figura 2.4: Número de ondas $k=4$, sobre uma malha fina com $N=12$ e sobre uma malha grossa com $N=6$ (Briggs et al., 2000)

O modo de Fourier antes de ser projetado da malha com 12 incógnitas para a malha com 6 incógnitas, tem $k=4$ e, pela definição, para $N=12$, tem-se um modo suave. Por outro lado, o modo de Fourier após ser projetado da malha fina para a malha grossa, tem $N=6$ para o mesmo $k=4$. Pela definição, tem-se um modo oscilatório. Pode-se concluir que o modo de Fourier, quando projetado para a malha grossa, torna-se um modo oscilatório.

Desta discussão pode-se concluir que, na malha mais fina, os erros de alta frequência são suavizados e, na malha mais grossa são suavizados os erros de alta e baixa frequência. Como o interesse é de se resolver problemas nas malhas mais refinadas possíveis, quando os

erros na malha fina apresentarem comportamento suave, recomenda-se a transferência do problema para uma malha com menos incógnitas (malha grossa). Nesta malha, as componentes de erro de baixa frequência serão suavizadas. Um esquema iterativo apropriado em diferentes níveis de malha dará uma rápida redução das componentes de alta frequência correspondentes e, como este processo passa por todas as frequências, uma rápida redução do erro global pode ser alcançada.

A taxa de convergência ideal (teórica) do método *multigrid* é independente do tamanho da malha, ou seja, independe do número de pontos da malha (ROACHE, 1998 e FERZIGER e PERIC, 2002). Segundo Briggs et al. (2000) para qualquer algoritmo *multigrid*, as mesmas componentes fundamentais são necessárias: deve haver uma sequência de malhas, operadores de transferência entre as malhas, um suavizador (*solver*) e operadores de malha grossa e fina. No algoritmo *multigrid* otimizado, tenta-se construir as componentes para os problemas e obter alta eficiência para o processo de solução. A idéia de um algoritmo *multigrid* robusto é escolher as componentes, independente do problema dado, para a maior classe de problemas possível (TROTTEMBERG et al., 2001).

Conforme Krechel e Stüben (1999), em qualquer abordagem com o método *multigrid*, suavização e correção na malha grossa são usadas em conjunto para eliminar o erro. Isto é devido às componentes do erro, que não podem ser corrigidas por um problema na malha grossa, que podem ser eficientemente reduzidas pela suavização. No método *multigrid* geométrico, uma hierarquia de malhas e interpolação linear são usadas, de forma que a relaxação pode ser escolhida de modo que o erro seja suavizado na idéia geométrica usual. Abordagens mais robustas empregam componentes sofisticadas do *multigrid*, como suavizadores complexos, interpolação operador-dependente, operadores de Galerkin e/ou malhas com múltiplos semi-engrossamentos. Uma alternativa para se obter a robustez é a abordagem destes problemas com o método *multigrid* algébrico.

A principal diferença entre *multigrid* algébrico e geométrico é que a aproximação geométrica emprega uma hierarquia de malhas fixa e uma interação entre suavização e correção da malha grossa é assegurada pela seleção apropriada do processo de suavização. Em contraste a isto, o AMG fixa o suavizador a algum esquema de relaxação simples, tal como relaxação Gauss-Seidel, e então é forçado a ter uma interação eficiente com a correção da malha grossa pela escolha do nível grosso e interpolação apropriadas (TROTTEMBERG et al., 2001).

De acordo com Krechel e Stüben (1999), esta forma de se tratar os problemas com o AMG, resultará em uma grande flexibilidade em adaptação das exigências específicas do

problema a ser resolvido. Esta flexibilidade irá fazer com que o AMG possa ser aplicado a vários problemas aos quais o *multigrid* geométrico não pode ser empregado.

2.3.2 Teoria básica do AMG

Inicialmente, no AMG, é feita a seleção de um esquema de relaxação, que permite determinar a natureza do erro suave. Como não se tem acesso à malha física do problema, a idéia de suavização pode ser definida algebricamente. Conforme Trottenberg et al. (2001), do ponto de vista algébrico, esta é uma questão importante para distinguir erros suaves de não suaves.

Definição 2.1: Erro suave, ou algebricamente suave, é qualquer erro que não é efetivamente reduzido por um método de relaxação (Briggs et al., 2000).

O próximo passo é o uso desta idéia de suavização para selecionar as malhas grossas, que serão subconjuntos das incógnitas. É necessário fazer a escolha do operador de transferência entre as malhas que permite engrossamento efetivo. De acordo com BRIGGS et al. (2000), depois de definido o esquema de relaxação, a malha grossa selecionada deve ser tal que as componentes suaves do erro sejam representadas acuradamente. O operador de interpolação deve ser definido de forma que as componentes suaves possam ser transferidas da malha grossa para a malha fina. Além disso, o operador de restrição e a versão da malha grossa de A devem ser definidos usando as propriedades variacionais.

Finalmente, são selecionadas as versões do operador da malha grossa, de forma que a correção da malha grossa tenha o mesmo efeito que no *multigrid* geométrico, ou seja, elimine as componentes de erro na variação do operador de interpolação.

Por questões didáticas, as componentes do AMG serão aqui descritas para dois níveis de malha, onde os índices h e H irão denotar malha fina e malha grossa, respectivamente. A Eq. (2.3) pode ser reescrita como:

$$A_h u^h = f^h \quad \text{ou} \quad \sum_{j \in \Omega^h} a_{ij}^h u_j^h = f_i^h \quad (i \in \Omega^h) \quad (2.17)$$

onde Ω^h denota o conjunto de índices $\{1, 2, \dots, n\}$. Implicitamente será assumido que A_h corresponde a uma matriz esparsa.

Para se obter os pontos nas malhas grossa e fina, é necessário fazer uma partição de Ω^h em dois subconjuntos disjuntos $\Omega^h = C^h \cup F^h$, onde C^h representa as variáveis que estão no nível grosso (variáveis C) e F^h é dado pelo conjunto complementar (variáveis F). Assumindo que tal partição é dada e definindo $\Omega^H = C^h$, o sistema AMG no nível grosso torna-se:

$$A_H u^H = f^H \quad \text{ou} \quad \sum_{l \in \Omega^H} a_{kl}^H u_l^H = f_k^H \quad (k \in \Omega^H) \quad (2.18)$$

que é construído baseado no princípio de Galerkin, ou seja, a matriz A_H é definida da seguinte forma:

$$A_H = I_H^H A_h I_H^h \quad (2.19)$$

onde I_H^h e I_h^H denotam os operadores de interpolação e restrição, respectivamente. Na subseção que trata de algoritmos (subseção 2.3.5), pode-se observar como é feita a construção da matriz na malha imediatamente mais grossa.

Como em qualquer método *multigrid*, é necessário um processo de suavização com um correspondente operador de suavização linear S_h . Ou seja, um passo de suavização é da forma:

$$\bar{u}^h = S_h u^h + (I_h - S_h) A_h^{-1} f^h \quad (2.20)$$

onde \bar{u}^h é a solução suavizada e I_h denota o operador identidade. Conseqüentemente, o erro $e^h = u^h - v^h$, onde u^h denota a solução exata da Eq.(2.17), é transformado como:

$$\bar{e}^h = S_h e^h \quad (2.21)$$

onde \bar{e}^h é o erro suavizado. Neste trabalho, o AMG será empregado com processos de suavização simples, como por exemplo, relaxação Gauss-Seidel.

A construção da partição C/F e dos operadores de transferência I_H^h e I_h^H envolvem processos proximamente relacionados e, sempre que forem mencionados os operadores de transferência, será assumido implicitamente que uma partição C/F é dada. De acordo com Trottenberg et al. (2001), estas componentes necessitam ser selecionadas como uma interface eficiente entre suavização e correção de malha grossa, para que a boa convergência seja obtida.

Uma observação importante deve ser feita de acordo com Trottenberg et al. (2001): as partições C/F devem ser criadas tão uniformes quanto possível, ou seja, com variáveis em F sendo “limitadas” pelas variáveis em C para interpolação. Embora não exista prova algébrica, a interpolação tende a ser consideravelmente melhor, resultando em convergência muito mais rápida.

Definição 2.2: Um ponto i é definido sendo (diretamente) acoplado (ou conectado) ao ponto $j \in \Omega^h$ se $a_{ij}^h \neq 0$.

Definição 2.3: A vizinhança para o ponto i é dada por:

$$N_i^h = \{j \in \Omega^h : j \neq i, a_{ij}^h \neq 0\} \quad (i \in \Omega^h) \quad (2.22)$$

O primeiro conceito importante para o *multigrid* algébrico já foi discutido: quando um erro é considerado algebricamente suave (Definição 2.1). Um segundo conceito a ser examinado é o de forte dependência ou forte influência. É necessário determinar que outras variáveis são mais importantes na i -ésima equação, ou seja, quais valores de u_j são mais importantes na i -ésima equação para se obter u_i . Para a determinação destas variáveis é necessário definir alguns conjuntos de pontos, conforme explicitado a seguir.

Definição 2.4: Uma variável i tem forte acoplamento negativo (ou está fortemente n -acoplada) a outra variável j , se

$$-a_{ij} \geq \theta \max_{a_{ik} < 0} |a_{ik}| \quad \text{com fixo } 0 < \theta < 1 \quad (2.23)$$

sendo θ aqui chamado de fator de redução da malha.

Definição 2.5: Denota-se S_i o conjunto de pontos que influenciam fortemente i , ou seja,

$$S_i = \{j \in N_i : i \text{ é fortemente } n\text{-acoplada a } j\} \quad (2.24)$$

Definição 2.6: O conjunto de pontos que dependem fortemente do ponto i é dado por:

$$S_i^T = \{j \in \Omega : i \in S_j\} \quad (2.25)$$

2.3.3 Engrossamento

Existem várias formas para se fazer o engrossamento no método *multigrid* algébrico disponíveis na literatura. Em Trottenberg et al. (2001) estão detalhados engrossamento padrão (baseado nas fortes conexões negativas), engrossamento agressivo e ainda engrossamento baseado nas fortes conexões positivas. Neste trabalho será implementado o engrossamento padrão, devido às entradas da matriz, que em geral, são negativas. Outro fator que influenciou nesta escolha é que o engrossamento agressivo apresenta duas desvantagens: faz com que a suavização seja menos efetiva e também com que a interpolação seja menos acurada (TROTTEMBERG et al., 2001).

Para o engrossamento padrão, considera-se que a partição C/F é baseada em acoplamentos diretos: para cada variável i em F é necessário ter um número mínimo de acoplamentos $j \in N_i$. Estes acoplamentos devem ser representados em C de forma que afetem o erro para o máximo em i . A partição C/F resultante será tal que todas as variáveis em F tenham uma substancial conectividade negativa (direta) para a vizinhança das variáveis em C . Com este procedimento, o engrossamento é essencialmente feito nas direções em que o erro algebricamente suave varia lentamente.

A proposta deste algoritmo corresponde a uma “escolha preliminar do ponto em C ”. O procedimento é iniciado quando é definida alguma variável i para se tornar uma variável em C . Todas as variáveis j , que são fortemente n -acopladas a i tornam-se variáveis em F . Na sequência, das variáveis indefinidas restantes (variáveis que ainda não foram tratadas), outra variável é definida para se tornar uma variável em C . Todas as variáveis indefinidas que são

fortemente n -acopladas a ela passam a ser variáveis em F . Este processo é repetido até que todas as variáveis sejam tratadas. O procedimento completo pode ser observado na Fig. 2.4.

Definição 2.7: Para qualquer variável indefinida i , é calculada uma “medida de importância”, λ_i :

$$\lambda_i = |S_i^T \cap U| + 2|S_i^T \cap F| \quad (i \in U) \quad (2.26)$$

onde U , para qualquer estágio do algoritmo, denota o conjunto de variáveis indefinidas, $|S_i^T \cap U|$ é a quantidade de variáveis indefinidas que dependem fortemente de i e $|S_i^T \cap F|$ é a quantidade de variáveis em F que dependem fortemente de i .

Para a obtenção de λ_i é necessário que exista uma tendência na construção da partição, iniciando em uma variável e seguindo até que todas sejam atingidas. O fator λ_i age como uma medida de quanto a variável $i \in U$ é uma variável em C , dando o estado atual de C e F . Para um estêncil de 9 pontos isotrópico, os passos do engrossamento são ilustrados na Fig. 2.5. Nessa figura, o conjunto Ω^h aparece no canto superior esquerdo e a seleção final na parte inferior central da figura. Os círculos representam os pontos, e os números que estão no seu interior, são os valores calculados para λ_i .

Os pontos que se encontram no contorno tornam-se variáveis em F . Desta forma, tais variáveis não requerem interpolação. Depois de terminado o algoritmo, todas as variáveis em F têm (pelo menos) um forte n -acoplamento a uma variável em C .

Na Fig. 2.5 (a), têm-se os valores de λ_i calculados para todos os pontos do domínio de cálculo. Na sequência, na Fig. 2.5 (b), toma-se o ponto que tem maior valor para λ_i , de acordo com ordenamento lexicográfico. Este ponto será um ponto em C . Os pontos que estão conectados a C , tornam-se pontos em F , conforme ilustrado na Fig. 2.5 (c). Na sequência, são recalculados os valores de λ_i para os pontos que faltam ser tratados, conforme Fig. 2.5 (d). O procedimento é repetido, considerando sempre ordenamento lexicográfico para a criação de um novo ponto em C . Para a construção da partição C/F o algoritmo a ser seguido está detalhado na subseção 2.3.5.

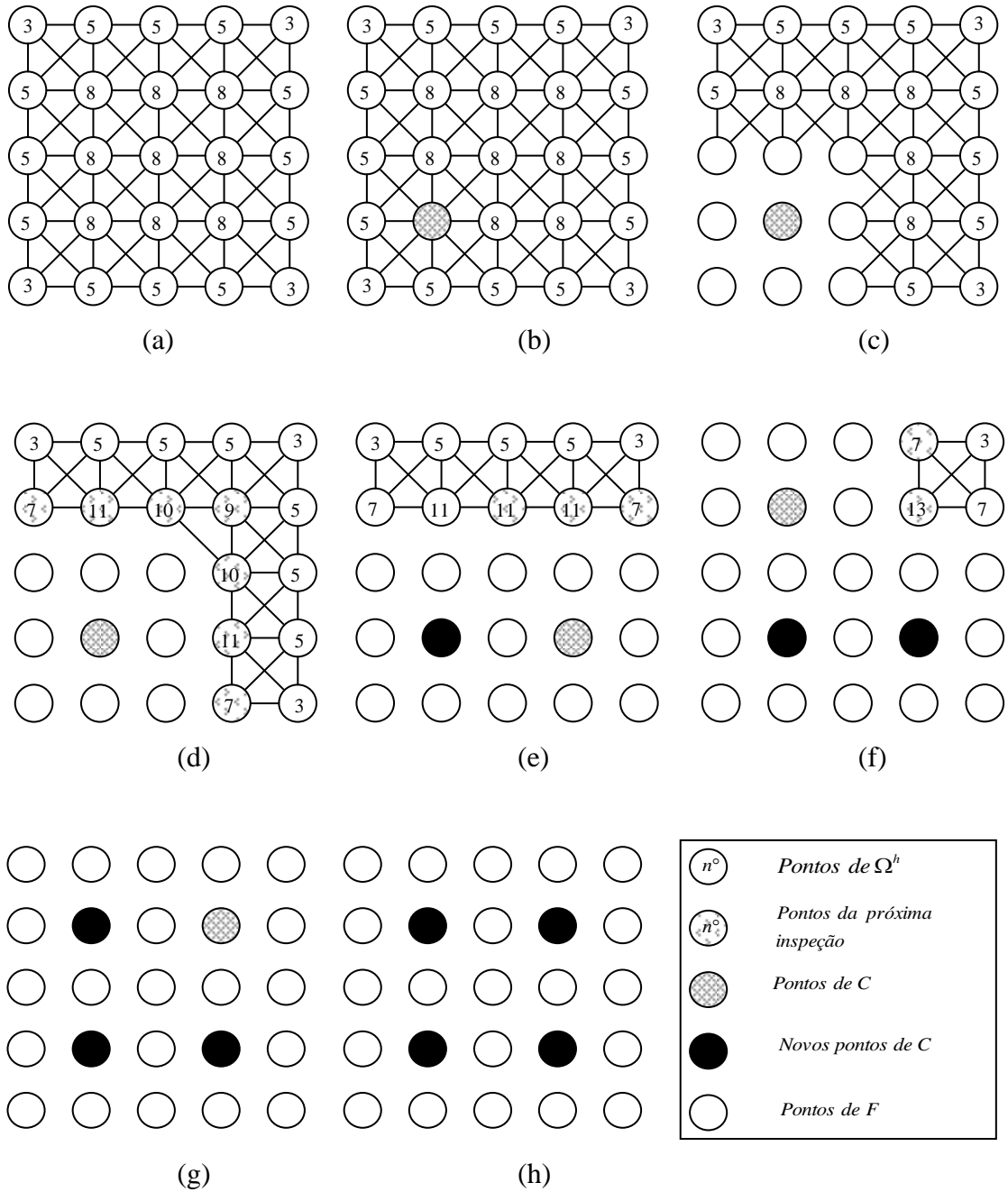


Figura 2.5: Sequência do procedimento de seleção dos pontos (Adaptado de Falgout, 2006)

2.3.4 Prolongação

Conforme Trottenberg et al. (2001) para se obter rápida convergência, erros algebricamente suaves necessitam ser bem aproximados pela interpolação. Por outro lado, a dimensão do operador do nível grosso (e o tempo para calculá-lo) depende fortemente do número de variáveis em C . Como a eficiência global é determinada pela velocidade de convergência e a quantidade de trabalho por ciclo, faz-se necessário limitar o número de

variáveis em C garantindo que todas as variáveis em F são fortemente conectadas às variáveis em C .

De acordo com Trottenberg et al. (2001), a prolongação no AMG pode ser obtida de várias formas: pelo método de interpolação direta, interpolação padrão, interpolação multipassos, interpolação de Jacobi ou ainda fazendo um truncamento da interpolação. Neste trabalho é utilizada a interpolação padrão, que será detalhada a seguir. Este procedimento é aplicado visto que a partição C/F é construída por meio de engrossamento padrão. Esta estratégia assegura que existe uma forte conectividade de $F-C$.

Supondo que e_i , $i \in C$, é um conjunto de valores na malha grossa representando um erro suave que pode ser interpolado para a malha fina $C \cup F$. Se um ponto j em C influencia fortemente um ponto i em F , então o valor e_j contribui fortemente para o valor de e_i na i -ésima equação (malha fina).

Conforme Briggs et al. (2000) uma “quantidade” u_i na malha fina pode ser interpolada por uma “quantidade” u_j na malha grossa se i depender fortemente de j . Isto é justificado pelo fato de que o erro suave varia lentamente na direção da forte conexão.

Desta forma, para cada ponto i na malha fina, é definida uma vizinhança de i , denotada por N_i , que já foi definida pela Eq. (2.22). Estes pontos podem ser divididos em três categorias (BRIGGS et al., 2000):

- os pontos da vizinhança da malha grossa que influenciam fortemente i , ou seja, o conjunto dos pontos interpoladores que estão na malha grossa, denotado por C_i , onde $C_i = C \cap S_i$;
- os pontos da vizinhança da malha fina que influenciam fortemente i , indicado por D_i^s , que é dado por $(D_i^s = D_i \cap S_i)$ onde $D_i = N_i - C_i$;
- os pontos que influenciam fracamente i , denotado por D_i^w . É um conjunto que pode conter pontos na malha grossa e na malha fina. Estes pontos são chamados de vizinhos fracamente conectados. O conjunto D_i^w é dado por $D_i^w = D_i - S_i$.

Os pontos que possuem forte e fraca influência sobre um ponto i podem ser observados na Fig. 2.6. Os pontos da malha grossa são indicados pelos círculos abertos. Os

pontos em C com forte influência sobre i são indicados com as linhas contínuas, os pontos em F que influenciam fortemente i com linhas tracejadas, e os pontos em F que influenciam fracamente i com linhas pontilhadas.

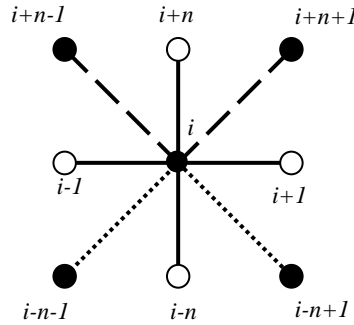


Figura 2.6: Ilustração de influências fortes e fracas sobre o ponto i (BRIGGS et al., 2000)

O operador de interpolação I_H^h é dado por:

$$(I_H^h e^H)_i = \begin{cases} e_i^H & \text{se } i \in C \\ \sum_{j \in C_i} w_{ij} e_j^H & \text{se } i \in F \end{cases} \quad (2.27)$$

onde w_{ij} são os pesos da interpolação dados pela Eq. (2.28), cuja dedução pode ser observada no Apêndice A:

$$w_{ij} = - \frac{a_{ij} + \sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in D_i^w} a_{in}} \quad (2.28)$$

Algumas variantes da interpolação descritas acima exploram os fortes acoplamentos em C , que aumentam “o raio” da interpolação, ou seja, aumentam a quantidade de pontos que podem ser empregados na interpolação. Em tais casos, é importante truncar de maneira adequada o operador de interpolação antes da obtenção do operador de Galerkin (TROTTERBERG et al., 2001). Sem esse truncamento, o operador de Galerkin pode ficar muito grande e o processo de suavização neste nível considerado mais grosso, pode ficar mais lento, além de utilizar mais memória computacional. Desta forma, antes de se obter o

operador de Galerkin na malha grossa, o operador de interpolação será truncado, ignorando as conexões que são menores que a maior delas (em valor absoluto), por uma constante, ε . Os pesos restantes são recalculados, de forma que a soma total permaneça inalterada.

Neste trabalho, esta constante é chamada de fator de forte dependência na malha grossa, sendo que sua aplicação está ligada à construção de mais um conjunto de dados (IWAMURA et al., 2003), conforme Eq. (2.29):

$$S_i^D = \left\{ j \in D_i^s : \sum_{l \in C_i} |a_{ij}| > \varepsilon \left(\frac{|a_{ij}|}{\max_{ik} |a_{ik}|} \right) \max_{jl} |a_{jl}| \quad \text{com } \varepsilon > 0 \text{ fixo} \right\} \quad (2.29)$$

O conjunto S_i^D denota os pontos fortemente dependentes do conjunto C_i . De acordo com Iwamura et al. (2003), um grande valor para ε aumenta a quantidade de pontos que estão na malha grossa, e isto irá aumentar o trabalho computacional. Na prática, é suficiente que um ponto tenha pelo menos uma forte conexão negativa.

2.3.5 Algoritmos

A aplicação do AMG a um dado problema é um processo composto de duas partes (KRECHEL e STÜBEN, 1999; TROTTENBERG et al., 2001):

- a fase inicial, totalmente automática, conhecida como fase *setup*. Nesta fase, são escolhidos os níveis grossos e definidos os operadores de malha grossa e de transferência entre as malhas;
- a fase de solução, que é direta, usa apenas as componentes já definidas na fase *setup*. O objetivo desta etapa é melhorar o processo do ciclo *multigrid* até que o nível de tolerância desejado seja atingido.

As componentes necessárias para criar um algoritmo para o AMG são (BRIGGS et al., 2000): esquema de relaxação, conjunto de pontos da malha grossa C , operador de malha grossa A_H , e os operadores de transferência entre malhas I_h^H e I_H^h . O algoritmo para fase *setup* do AMG é dado na Tab. 2.2:

Tabela 2.2: Algoritmo para fase *setup* do AMG (Adaptado de Cleary et al., 2000)

-
- 1) Partição Ω^h em conjuntos disjuntos C^h e F^h ;
 - a) Faça $\Omega^H = C^h$;
 - b) Defina a interpolação I_H^h ;
 - 2) Faça $I_h^H = (I_H^h)^T$ e $A_H = I_h^H A_h I_H^h$;
 - 3) Se Ω^H for pequeno o bastante pare. Caso contrário, faça $\Omega^h = \Omega^H$ e volte ao passo 1.
-

Para a partição de Ω^h são empregados dois algoritmos consecutivamente: o primeiro detalha a escolha preliminar do ponto em C (Tab. 2.3) e o segundo faz a escolha final do ponto em C e define os pesos da interpolação (Tab. 2.4). Estes dois algoritmos estão detalhados a seguir:

Tabela 2.3: Escolha preliminar do ponto em C (Ruge e Stüben, 1986)

-
- 1) Seja $C = \{ \}$, $F = \{ \}$, $U = \Omega^h$, e $\lambda_i = |S_i^T \cap U| + 2|S_i^T \cap F|$, para todo i .
 - 2) Tome $i \in U$ com máximo λ_i . Seja $C = C \cup \{i\}$ e $U = U - \{i\}$.
 - 3) Para todo $j \in S_i^T \cap U$, faça os passos 4 e 5.
 - 4) Seja $F = F \cup \{j\}$ e $U = U - \{j\}$.
 - 5) Para todo $l \in S_j \cap U$, faça $\lambda_l = \lambda_l + 1$.
 - 6) Para todo $j \in S_i \cap U$ faça $\lambda_j = \lambda_j - 1$.
 - 7) Se $U = \{ \}$, pare. Caso contrário, volte ao passo 2.
-

Tabela 2.4: Escolha final do ponto em C e definição dos pesos da interpolação (Adaptado de Iwamura et al., 2003)

-
- 1) Seja $T = \{ \}$.
 - 2) Se $T \supseteq F$, pare. Caso contrário, tome $i \in F - T$ e $T = T \cup \{i\}$.
 - 3) Sejam C_i , D_i^S , D_i^W , S_i^D e o conjunto $\overline{C}_i = \{ \}$.
 - 4) Seja $d_{ii} = a_{ii}$ e para $k \in C_i$, $d_k = a_{ik}$.
 - 5) Para cada $j \in D_i^W$, $d_i = d_i + a_{ij}$.
 - 6) Para cada $j \in D_i^S$,
 - se $j \in S_i^D$
então $d_k = d_k + a_{ij} a_{jk} / \sum_{l \in C_i} a_{jl}$ para $k \in C_i$,
 - senão,
se $\overline{C}_i \neq \{ \}$
seja $C = C \cup \{i\}$ e $F = F - \{i\}$ e volte ao passo 2.
 - senão $\overline{C}_i = \{j\}$, $C_i = C_i \cup \{j\}$, $D_i^S = D_i^S - \{j\}$, atualize S_i^D e volte ao passo 4.
 - 7) Seja $C = C \cup \overline{C}_i$, $F = F - \overline{C}_i$ e $w_{ik} = -d_k / d_i$ para cada $k \in C_i$ e volte ao passo 2.
-

Haase e Langer (2002) trazem um algoritmo para a geração da matriz de coeficientes na malha imediatamente mais grossa, que pode ser obtida a partir da Eq.(2.30):

$$[A_H]_{ij} = \left[\sum_{k \in N_i} \sum_{l \in N_j} \alpha_{ki} A_{kl} \alpha_{lj} \right]_{ij} \quad (2.30)$$

Para aplicá-lo é necessário definir alguns conjuntos, obtidos depois que a fase *setup* é concluída. Estes conjuntos são montados com base na partição C/F . Para cada ponto de Ω^h , é determinado o conjunto dos “vizinhos grossos” e o conjunto dos “vizinhos finos”, detalhados a seguir:

$$C^i = \{j \in C / a_{ij} \neq 0, i \in F\} \quad (2.31)$$

$$F^i = \{j \in F / a_{ij} \neq 0, i \in F\} \quad (2.32)$$

Os pesos da interpolação podem ser calculados por:

$$\alpha_{ij} = \begin{cases} 1 & i = j \in C \\ -\frac{(A_{ij} + c_{ij})}{A_{ii} + c_{ii}} & i \in F, j \in C^i \\ 0 & \text{caso contrário} \end{cases} \quad (2.33)$$

onde, A_{ij} são as entradas da matriz de coeficientes (da malha fina) e c_{ij} são os coeficientes dados por:

$$c_{ij} = \sum_{k \in F^i} \frac{A_{ik} \cdot A_{kj}}{\sum_{l \in C^i} A_{kl} + A_{ki}} \quad (2.34)$$

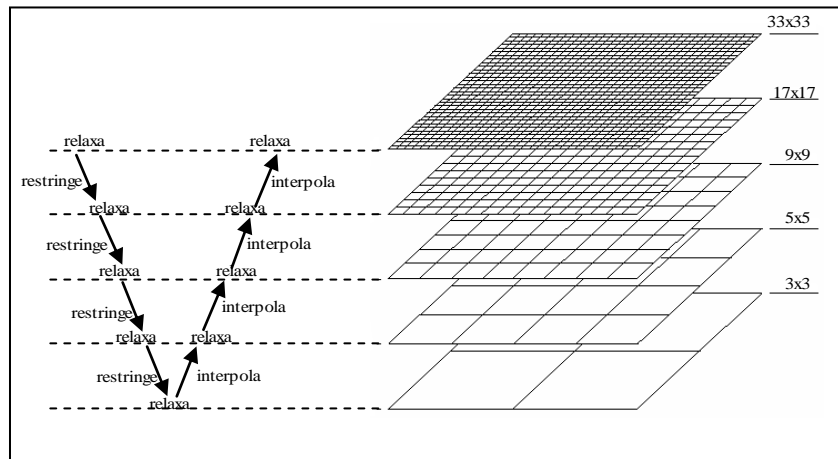
O algoritmo para gerar a matriz de coeficientes na malha mais grossa pode ser observado na Tab. 2.5:

Tabela 2.5: Algoritmo para gerar a matriz de coeficientes (HAASE e LANGER, 2002)

- 1) $A_H = 0$;
- 2) Para todo $k \in F \cup C$ faça:
- 3) Determine F^k e C^k ;
- 4) Para todo $l \in F^k \cup C^k \cup \{k\}$ faça:
- 5) $T = C^k \cup C^l \cup (\{l\} \cap C)$;
- 6) Para todo $i \in T$ faça:
- 7) Para todo $j \in T$ faça:
- 8) $A_{H,i,j} = A_{H,i,j} + \alpha_{ki} A_{kl} \alpha_{lj}$

Depois das componentes algorítmicas $(\Omega^h, \Omega^H, I_h^H, I_H^h, A_h, A_H)$ estarem previamente definidas, um algoritmo de dois níveis pode ser montado. Este algoritmo, aplicado recursivamente para outros níveis, dará origem ao ciclo *multigrid*.

Na Fig. 2.7 pode-se observar a ilustração para uma malha estruturada quadrangular, onde estão descritos os procedimentos a serem realizados em cada nível de malha. Ao lado do ciclo, são mostradas as malhas utilizadas para um ciclo V, iniciando com uma malha de 33x33 nós até uma malha de 3x3 (a malha mais grossa). Esta forma de se visitar as malhas, que é utilizada para o método *multigrid* geométrico, pode ser aplicada de maneira análoga ao AMG.

**Figura 2.7: Ciclo V (Adaptado de Craig, 2007)**

Outros tipos de ciclos (W, F, entre outros) podem ser obtidos por analogia ao método *multigrid* geométrico. De acordo com Ruge e Stüben (1986) e Trottenberg et al. (2001), a forma mais usada para superar a convergência h -dependente (dependente da malha), é o uso de ciclos “melhores” como os ciclos W ou F. Porém, além do fato destes ciclos serem mais custosos (fato que pode ser considerável em AMG dependendo do engrossamento utilizado), eles terão o mesmo fator de convergência que um método dois níveis correspondente, o que

pode levar à convergência h -dependente. Devido a estes motivos, neste trabalho é empregado o ciclo V.

O algoritmo CS para a aplicação do AMG em dois níveis (ciclo V) pode ser observado na Tab. 2.6.

Tabela 2.6: Algoritmo CS para o AMG com duas malhas (Adaptado de BRIGGS et al., 2000)

$$v^h \leftarrow AMG(v^h, f^h)$$

- 1) Iterar η vezes $A_h u^h = f^h$ com estimativa inicial v_0^h ;
- 2) Calcular o resíduo em Ω^h : $r^h = f^h - A_h v^h$;
- 3) Restringir o resíduo para Ω^H : $r^H = I_h^H r^h$;
- 4) Resolver $A_H e^H = r^H$ em Ω^H , com estimativa inicial $v_0^H = 0$;
- 5) Interpolar o erro para Ω^h : $e^h = I_H^h e^H$;
- 6) Corrigir a aproximação na malha fina $v^h \leftarrow v^h + e^h$;
- 7) Iterar η vezes $A_h u^h = f^h$ com estimativa inicial v^h .

As aproximações específicas para dois níveis definidas acima podem ser estendidas para diversos níveis de maneira direta, aplicando recursivamente a mesma estratégia. O ciclo V para mais malhas é obtido substituindo a solução direta do problema na malha grossa com uma chamada recursiva do AMG em todas as malhas. Assumindo que as equações no nível mais grosso são resolvidas exatamente, o ciclo V resultante converge em apenas um passo de iteração (TROTTEBERG et al., 2001).

2.4 Modelos matemáticos e numéricos

Para se alcançar os objetivos propostos neste trabalho, são resolvidos três problemas com o uso do método *multigrid* algébrico, todos com soluções analíticas conhecidas: equação de Laplace, considerando a temperatura como variável de interesse, com condição de contorno dada por uma função linear ao norte e a leste (problema linear); equação de Laplace com condição de contorno dada por uma função senoidal ao norte (problema senoidal) e equação de Poisson. Todas são consideradas em um domínio de cálculo quadrado, dado por $\{(x, y) \in \mathbb{R}^2 : 0 \leq x, y \leq 1\}$, que é discretizado tanto com o uso de malhas estruturadas quadrangulares quanto de malhas estruturadas triangulares. Esta última forma de discretização não é aplicada à equação de Poisson. Para a discretização das equações, em malhas

quadrangulares é empregado o método de diferenças finitas e em malhas triangulares, o método dos volumes finitos.

A equação de Laplace com condição de contorno dada por uma função linear é dada por:

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, & 0 < x, y < 1 \\ T(x, 0) = T(0, y) = 0, T(x, 1) = x, T(1, y) = y \end{cases} \quad (2.35)$$

Com solução analítica dada por:

$$T(x, y) = x y \quad (2.36)$$

A equação de Laplace, com condição de contorno dada por uma função senoidal é dada por:

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, & 0 < x, y < 1 \\ T(x, 0) = T(0, y) = T(1, y) = 0, T(x, 1) = \text{sen}(\pi x) \end{cases} \quad (2.37)$$

Com solução analítica dada por:

$$T(x, y) = \text{sen}(\pi x) \frac{\text{senh}(\pi y)}{\text{senh}(\pi)} \quad (2.38)$$

A equação de Poisson é dada por:

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = S, & 0 < x, y < 1 \\ T(x, 0) = T(0, y) = T(1, y) = T(x, 1) = 0 \end{cases} \quad (2.39)$$

Com termo fonte dado por:

$$S = -2[(1 - 6x^2)y^2(1 - y^2) + (1 - 6y^2)x^2(1 - x^2)] \quad (2.40)$$

E solução analítica dada por:

$$T(x, y) = (x^2 - x^4)(y^4 - y^2) \quad (2.41)$$

Os dois problemas resolvidos a partir da equação de Laplace são resolvidos em Oliveira et al. (2008), onde se tem resultados a respeito de parâmetros ótimos (número de iterações internas e número de níveis) para o método *multigrid* geométrico. Além disso, a equação de Poisson é a mesma empregada em Briggs et al. (2000), que a resolve em um problema teste para uma comparação entre AMG e GMG.

O código computacional empregado para a obtenção dos resultados aqui apresentados teve como base o programa AMG1R6, de Ruge e Stüben (1986). Deste código é utilizada a fase *setup*, que emprega engrossamento padrão (baseado nas fortes conexões negativas) e prolongação padrão (implementada devido ao uso do engrossamento padrão). Para a fase de solução é empregado ciclo V, estimativa inicial nula, *solver* Gauss-Seidel lexicográfico, critério de parada baseado na norma l_2 do resíduo e tolerância de 10^{-8} . No código original de Ruge e Stüben foram feitas diversas modificações na fase de solução, para atender aos objetivos do presente trabalho. A forma de entrada dos dados também foi modificada, de forma que, para cada tipo de malha (quadrangular ou triangular) existe um gerador de matrizes específico, acoplado ao programa principal. O gerador para malhas triangulares é de autoria de Araki e Alves (2009), que o empregam para a obtenção de soluções numéricas com o objetivo de fazer análise de erros numéricos em malhas triangulares.

A máquina empregada para a obtenção dos resultados desta tese é o CFD-9, que conta com processador Intel Core2Quad e 8 Gb de memória, que é utilizada de forma total devido ao uso do sistema operacional 64bits. A linguagem empregada é Fortran 2003, com compilador Intel Fortran 9.1.

O código obtido para a resolução dos problemas com o emprego do AMG é verificado através da análise de erros numéricos, conforme detalhado no Capítulo 3. Para esta análise, o comportamento do erro numérico e de suas ordens é avaliado para algumas variáveis de interesse: temperatura no ponto central do domínio de cálculo, temperatura média (obtida pelas regras do retângulo ou do trapézio, de acordo com o tipo de malha empregado na discretização), norma l_∞ (norma infinito) e norma \bar{l}_1 , ambas do erro numérico. Estas variáveis de interesse são detalhadas na sequência.

Para a obtenção da temperatura no ponto central do domínio de cálculo é feita a substituição das coordenadas $x=0,5$ e $y=0,5$, nas soluções analíticas dadas pelas Eqs. (2.36), (2.38) e (2.41). A temperatura média é obtida pela seguinte equação:

$$T_m = \int_0^1 \int_0^1 T(x, y) dx dy \quad (2.42)$$

onde $T(x, y)$ é a solução analítica para cada equação.

Outra variável de interesse considerada para a verificação do código computacional é o erro numérico dado pela Eq. (2.43), medido pela norma l_∞ do erro numérico, dada pela Eq. (2.9), repetida aqui na Eq. (2.44).

$$E(i) = \Phi(i) - \phi(i) \quad (2.43)$$

$$l_\infty = \|E\|_\infty = \max_{1 \leq i \leq N} |E(i)| \quad (2.44)$$

onde Φ representa a solução analítica, ϕ é a solução numérica e N é o número de incógnitas. O valor para esta variável de interesse deve tender à zero com o refino da malha. A última variável de interesse avaliada é o erro numérico, sendo agora medido pela norma \bar{l}_1 do erro, dada pela Eq. (2.45). O valor analítico para esta variável de interesse também tende a zero.

$$\bar{l}_1 = \frac{\sum_{i=i_0}^{i_f} |\Phi(i) - \phi(i)|}{N} \quad (2.45)$$

onde i_0 e i_f são os valores para i que variam conforme o tipo de malha (método empregado na discretização) utilizada.

De acordo com Fortuna (2000), para tratar o modelo numérico computacionalmente, é necessário expressar de forma adequada as equações e a região (domínio de cálculo) onde elas são válidas. Como não é possível obter soluções numéricas sobre uma região contínua, devido aos infinitos pontos da mesma, inicialmente o domínio de cálculo é discretizado, isto é, dividido em pontos (ou volumes). Nesses pontos (ou volumes) as soluções são obtidas, sendo

que ao conjunto dos pontos (ou volumes) discretos dá-se o nome de malha. A distribuição adequada dos pontos (ou volumes) no domínio de cálculo é fundamental para se obter uma solução numérica representativa do problema a ser resolvido.

Na seqüência são apresentados os métodos numéricos utilizados na discretização dos modelos matemáticos, onde os termos das equações são escritos em função dos valores das incógnitas em pontos discretos adjacentes. O resultado é um conjunto de equações algébricas, geralmente lineares, que podem ou não estar acopladas. Nesta etapa, são introduzidas as condições de contorno do problema, normalmente modificando-se apropriadamente as equações para pontos perto das fronteiras. As condições de contorno, juntamente com as propriedades físicas do fluido, e os parâmetros do escoamento, especificam o problema a ser tratado.

2.4.1 Formulação para o método de diferenças finitas

O princípio fundamental do método de diferenças finitas (MDF) é aproximar através de expressões algébricas, cada termo do modelo matemático em cada nó da malha. Segundo Ferziger e Peric (2002), o ponto inicial é a equação da conservação na forma diferencial. O domínio de solução é coberto por uma malha, conforme ilustrado na Fig. 2.8. Para cada ponto desta malha, a equação diferencial é aproximada pela substituição das derivadas parciais por aproximações em termos dos valores nodais das funções. O resultado é uma equação algébrica por nó (ponto) da malha, em que o valor de cada variável e de um determinado número de vizinhos aparecem como incógnitas.

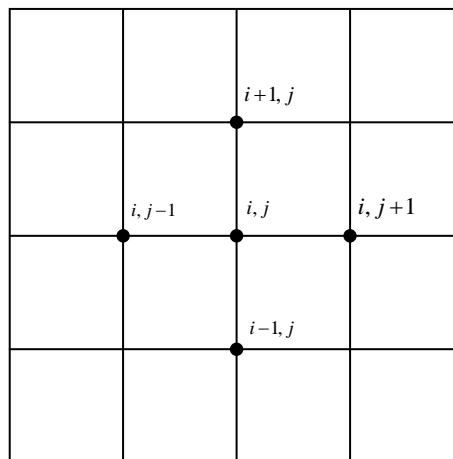


Figura 2.8: Discretização com diferenças finitas

Expansões em série de Taylor ou ajuste polinomial são usados para obter as aproximações para a primeira e segunda derivadas das variáveis com respeito às coordenadas. Neste trabalho são aplicadas as aproximações obtidas das expansões em série de Taylor. Para a equação de Laplace e Poisson são aplicadas aproximações do tipo CDS-2 (*Central Differencing Scheme*) (Ferziger e Peric, 2002), com relação a x e a y , que são dadas pelas Eqs. (2.46) e (2.47), respectivamente.

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{\Delta x^2} \quad (2.46)$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{\Delta y^2} \quad (2.47)$$

onde $\Delta x = \frac{1}{N_x - 1}$, $\Delta y = \frac{1}{N_y - 1}$, com N_x e N_y denotando a quantidade pontos nas direções x e y , respectivamente. Estas aproximações podem ser substituídas na equação de Laplace, Eq. (2.35). Depois de rearranjados os termos, chega-se à seguinte equação:

$$\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right) T_{i,j} - \frac{1}{\Delta x^2} T_{i,j-1} - \frac{1}{\Delta x^2} T_{i,j+1} - \frac{1}{\Delta y^2} T_{i-1,j} - \frac{1}{\Delta y^2} T_{i+1,j} = 0 \quad (2.48)$$

A Eq. (2.48) é válida para todos os pontos internos, sendo que, as condições de contorno, que são dadas por temperaturas conhecidas, são transferidas para o termo fonte dos pontos internos. Desta forma, não se tem equações para as variáveis que estão nos contornos.

A temperatura média, que é calculada com a regra do trapézio, é dada por:

$$T_m = \frac{\Delta x \Delta y}{4} \sum_{i=2}^{N_x} \sum_{j=2}^{N_y} (T_{i,j} + T_{i,j-1} + T_{i-1,j-1} + T_{i-1,j}) \quad (2.49)$$

2.4.2 Formulação para o método dos volumes finitos

Para o método dos volumes finitos (MVF) não interessam a forma e o modo como foi criado o volume elementar. A característica básica deste método é a integração das equações, na forma conservativa, sobre um volume elementar qualquer (MALISKA, 2004).

De acordo com Ferziger e Peric (2002), o método dos volumes finitos pode ser aplicado a qualquer tipo de malha, sendo aceitável também para geometrias complexas. As malhas definem somente os contornos do volume de controle e não necessitam ser relacionadas ao sistema de coordenadas. O método é conservativo por construção, assim as integrais de superfície (que representam os fluxos difusivos e advectivos) são as mesmas para os volumes de controle que compartilham os contornos.

No método dos volumes finitos, o elemento é definido como sendo a entidade geométrica delimitada pelos nós da malha, e o volume de controle é a região do domínio de cálculo onde as equações diferenciais governantes são integradas. Nas formulações em que o volume de controle é escolhido como sendo o próprio elemento, e as variáveis de interesse ficam armazenadas no centro do volume de controle (ou do elemento) são chamadas *cell center*, pois o centro do volume de controle coincide com o centro do elemento (MALISKA, 2004). Esta formulação está ilustrada na Fig. 2.9.

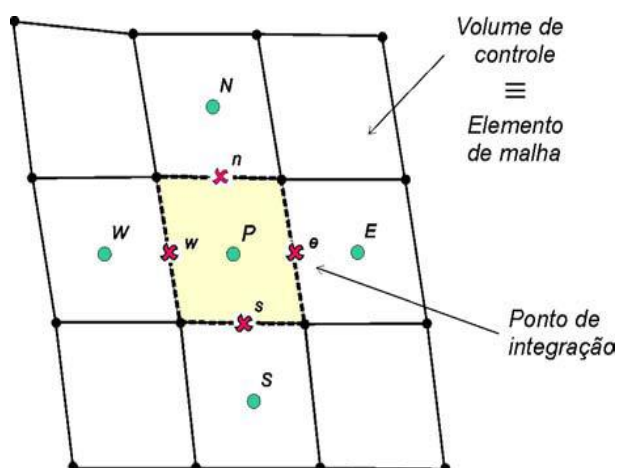


Figura 2.9: Malha estruturada - volume coincidente com o elemento (CORDAZZO, 2006)

As vantagens de uma abordagem baseada em volumes de controle estão listadas em Minkowycz et al. (1988): o esquema resultante da discretização pode ser considerado estritamente conservativo. O fato da discretização das equações governantes ser baseada no balanço do escoamento significa que os termos nas equações algébricas resultantes têm uma interpretação física específica. A implementação do processo de discretização baseada no volume de controle é direta e tem custo reduzido.

Na discretização do modelo matemático são necessárias expressões algébricas para a variável dependente e sua derivada de primeira ordem, avaliadas nas faces dos volumes de controle. Essas expressões são chamadas funções de interpolação. Em geral, estas funções têm o objetivo de avaliar o valor de uma propriedade genérica ϕ na interface do volume de controle bem como suas derivadas.

Tradicionalmente, o MVF utiliza dois tipos de aproximação (MALISKA, 2004): um para a determinação dos valores das derivadas de ϕ e outro para a determinação dos valores de ϕ nas interfaces de integração. Normalmente, na avaliação das derivadas é suficiente o uso de um esquema de diferenças centrais, e na avaliação de ϕ , é empregado algum mecanismo que considere os efeitos advectivos do problema.

Existem duas formulações, que podem ser empregadas, para a construção dos volumes de controle usando elementos triangulares (Maliska, 2004): triangulação de Delaunay (por diagramas de Voronoi) e método das medianas, sendo que neste trabalho, é aplicado o método das medianas. Para esta formulação, os volumes são gerados por uma triangulação que fornece as coordenadas dos vértices dos volumes. Na Fig. 2.10 pode ser observado um volume de controle que coincide com o elemento em uma malha triangular estruturada. Nesta figura, os elementos 1, 2, 3 e 4 são os volumes considerados reais e os volumes 5, 6, 7 e 8 são os volumes fictícios, utilizados para a aplicação das condições de contorno.

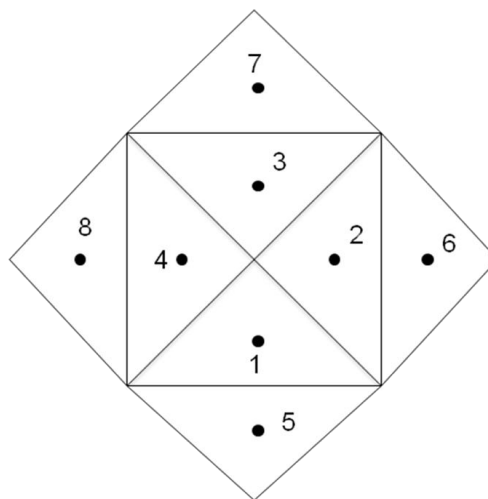


Figura 2.10: Volume de controle para uma malha triangular

A equação diferencial parcial na forma conservativa é integrada, já que o MVF cria as equações aproximadas através do balanço de conservação da propriedade envolvida (massa, quantidade de movimento, entalpia, etc.). Os volumes de integração (elementos 1, 2, 3 e 4)

podem ser observados na Fig. 2.10. As equações de conservação aproximadas para os volumes de controle são obtidas através da montagem elemento-por-elemento. Nesta figura, os pontos localizados no centro dos triângulos representam os centróides do volume de controle.

O método dos volumes finitos é ilustrado pela aplicação do método à equação de Laplace bidimensional, escrita abaixo na sua forma geral:

$$\nabla \cdot (\Gamma^\phi \nabla \phi) = 0 \quad (2.50)$$

que pode ser reescrita como:

$$\nabla \cdot \vec{J} = 0 \quad (2.51)$$

onde $\vec{J} = \Gamma^\phi \nabla \phi$ é o chamado fluxo difusivo, o operador nabla (∇) dado por $\nabla(\) = \frac{\partial(\)}{\partial x} \vec{i} + \frac{\partial(\)}{\partial y} \vec{j}$, ϕ é a propriedade envolvida (no caso a temperatura) e Γ^ϕ é o coeficiente de difusão (no caso a condutividade térmica). Integrando a Eq. (2.51) em cada volume de controle, tem-se:

$$\int_V \nabla \cdot \vec{J} dV = 0 \quad (2.52)$$

Com a aplicação do Teorema da Divergência de Gauss (GREENBERG, 1998), a integral de volume será transformada em uma integral de superfície, ou seja:

$$\int_S (\vec{J} \cdot \vec{n}) ds = 0 \quad (2.53)$$

Esta integral de superfície pode ser reescrita como um somatório, passando o problema de contínuo para discreto:

$$\sum \vec{J} \cdot \vec{n} ds = 0 \quad (2.54)$$

O produto escalar indicado na Eq. (2.54) resulta na derivada direcional. A equação discretizada é então dada por:

$$\sum \left(\Gamma^\phi \frac{\partial \phi}{\partial \vec{n}} \Delta s \right)_{p_i} = 0 \quad (2.55)$$

onde p_i indica o ponto de integração da malha. Esta equação discretizada pode ser aplicada a qualquer volume de controle e pode ser reescrita como:

$$\sum_i \Gamma^\phi \left(\frac{\partial \phi}{\partial \vec{n}} \right)_i \Delta s_i = 0 \quad (2.56)$$

Na Eq. (2.56) as derivadas são aproximadas pelas fórmulas de interpolação empregadas na discretização com volumes finitos. O sistema de equações resultante da discretização do domínio de cálculo é dado por:

$$A_i \phi_i = \sum_{vizinhos} A_{NB} \phi_{NB} + B_i \quad (2.57)$$

onde NB denota cada um dos vizinhos do ponto i em questão. Na Fig. 2.9, por exemplo, para $i=1$, tem-se que $NB = 2, 4$ e 5 .

As condições de contorno são aplicadas com o uso de volumes fictícios (ilustrados na Fig. 2.11). Este método é empregado, pois são utilizados volumes inteiros para todos os volumes, respeitando assim, os princípios de conservação para todo o domínio. Todos os volumes do domínio, inclusive os de fronteira, são interpretados como internos, uma vez que são criados os volumes fictícios. A desvantagem é a criação de novas incógnitas, aumentando o tamanho do sistema linear, situação que vai se agravando quando a dimensão do problema aumenta.

Para condição de contorno de Dirichlet, a propriedade ϕ na fronteira é conhecida, ϕ_f , e pode ser escrita como:

$$\phi_f = \frac{\phi_P + \phi_E}{2} \quad (2.58)$$

A Eq. (2.58) determina os coeficientes dos volumes fictícios para a Eq. (2.57).

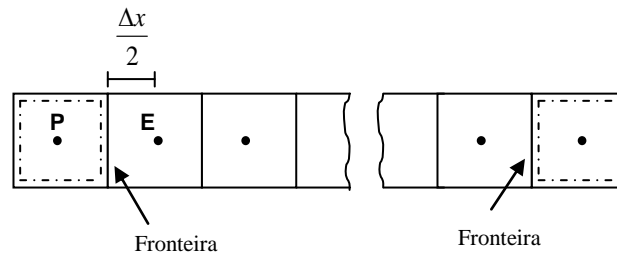


Figura 2.11: Condições de contorno com volumes fictícios (MALISKA, 2004)

Para o método dos volumes finitos, a variável de interesse temperatura média, é obtida com a aplicação da regra do retângulo, dada pela Eq. (2.59):

$$T_m = \frac{\sum_{i=1}^N T_i * \text{área_volume}}{\text{área_total}} \quad (2.59)$$

onde *área_volume* é a área do volume de controle que é usado no cálculo e *área_total* é a área do domínio de cálculo.

3 Verificação do código computacional

Este capítulo é destinado à análise dos códigos computacionais empregados na obtenção dos resultados deste trabalho. Esta análise é feita através do estudo de erros numéricos. São analisadas as equações de Laplace, em malhas triangulares e quadrangulares, e a equação de Poisson em malhas quadrangulares. A tolerância a ser empregada nos testes computacionais também é determinada com a ajuda dos testes computacionais aqui explicitados. O principal objetivo deste capítulo é mostrar a coerência dos códigos computacionais empregados.

A origem dos erros está intimamente ligada aos processos de análise e solução de um problema. Há dois grandes grupos de métodos para a análise e solução de problemas: os experimentais e os teóricos. Os teóricos, por sua vez, podem ser divididos em dois grupos distintos: analíticos e numéricos. Na Fig. 3.1 pode ser observada esta divisão entre os métodos de solução e os respectivos erros gerados em cada fase. Nessa figura, as indicações existentes entre as soluções numérica e analítica não indicam dependência entre elas, apenas ilustram os erros que estão associados a cada fase.

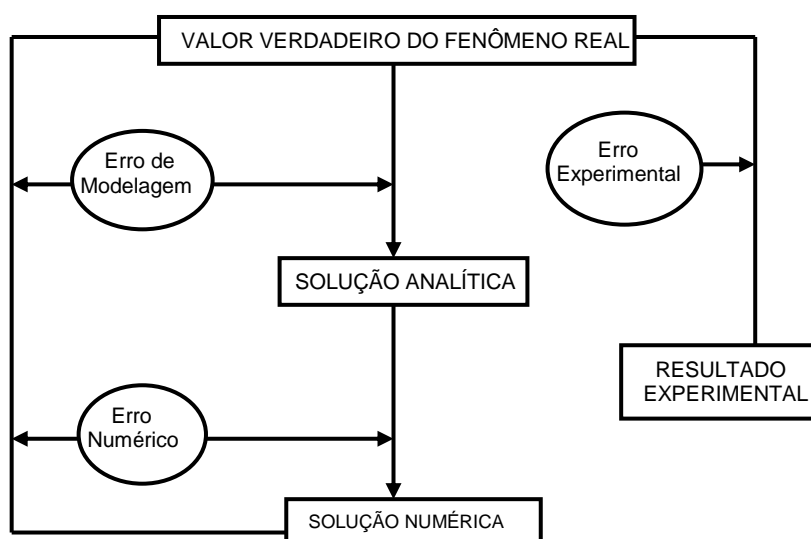


Figura 3.1: Erros envolvidos nos métodos de engenharia (Marchi, 2001)

Como este trabalho trata de soluções numéricas de equações diferenciais parciais, os erros experimentais, mostrados na Fig. 3.1, não são descritos aqui, uma vez que estão sendo considerados apenas métodos de soluções numéricas de problemas, ou seja, métodos teóricos.

A magnitude aceitável para o erro numérico depende, dentre outros fatores, da finalidade da solução numérica, dos recursos financeiros envolvidos, do tempo permitido para realizar as simulações e dos recursos computacionais disponíveis. Sabendo-se que as soluções numéricas contêm erros, é importante estimá-los pelos seguintes motivos (MARCHI, 2001):

- Quando o erro é maior do que o aceitável, compromete-se a confiabilidade do uso da solução numérica;
- Quando o erro é menor do que o necessário, há desperdício de recursos computacionais, isto é, de tempo de CPU e de quantidade de memória;
- Para validar e desenvolver modelos matemáticos que visem explicar fenômenos físico-químicos ainda não modelados adequadamente e cujas soluções analíticas são desconhecidas; um exemplo típico é a modelagem de escoamentos turbulentos;
- Para otimizar o uso da malha, isto é, adaptá-la visando homogeneizar o nível de erro no domínio de cálculo; e
- Para evitar interpretações equivocadas.

Nesta etapa do trabalho, os erros de iteração, arredondamento e programação são minimizados e considerados desprezíveis. Os erros de iteração são minimizados, pois para estas análises, o processo iterativo é levado até o erro de máquina ser alcançado (em torno de treze casas decimais). Os erros de arredondamento são minimizados com o emprego de precisão dupla nos cálculos das variáveis de interesse, e os erros de programação são eliminados através da verificação das soluções numéricas. Desta forma, considera-se que o problema em questão tem a influência do erro de truncamento, que, nestas circunstâncias, é chamado de erro de discretização (Marchi, 2001).

Conforme exposto no Capítulo 2, o erro a ser considerado aqui é o erro de discretização, repetido aqui por fins didáticos:

$$E(\phi) = \Phi - \phi \quad (3.1)$$

onde Φ é a solução analítica da variável de interesse e ϕ é a solução numérica. As estimativas deste erro podem ser divididas em dois tipos básicos (Szabó e Babuska, 1991): estimativas *a priori* ou *a posteriori* da solução numérica:

- Estimativas *a priori*: Proporcionam uma análise qualitativa do erro de discretização antes mesmo de se obter uma solução numérica. O objetivo de uma estimativa *a priori* é obter a ordem assintótica da equação diferencial discretizada. Com esta estimativa, antes de se obter qualquer solução numérica, é possível prever o comportamento assintótico do erro de discretização com relação ao refinamento da malha e à ordem aparente. Também é possível avaliar qual é o efeito da redução do tamanho dos elementos da malha sobre o erro de discretização (E) da solução numérica.
- Estimativas de erro *a posteriori*: As estimativas de erro *a posteriori* são usadas para estimar a magnitude do erro de discretização. Existem vários métodos que podem ser empregados. Eles podem ser divididos em dois grandes conjuntos. No primeiro, as estimativas de erro são baseadas na solução numérica obtida numa única malha. No segundo conjunto, as estimativas de erro são baseadas nas soluções numéricas obtidas em malhas múltiplas, em geral, os métodos de diferenças finitas e dos volumes finitos se enquadram neste conjunto. Alguns estimadores deste tipo são: delta, Richardson, GCI, multicoeficientes e convergente (MARCHI, 2001).

Neste trabalho são analisadas as ordens efetiva e aparente. Para a obtenção da ordem efetiva é necessário o erro verdadeiro (solução analítica menos a solução numérica) em duas malhas. Para a obtenção da ordem aparente são necessárias três soluções numéricas. Estas duas ordens permitem verificar *a posteriori* se a ordem assintótica p_L , obtida da discretização das equações diferenciais, ou seja, obtida *a priori* da solução numérica, é atingida. Será analisada também a queda do erro numérico (diferença entre solução analítica e numérica), conforme a malha é refinada.

A ordem efetiva, obtida com base na solução analítica, permite verificar se à medida que o tamanho do volume de controle é reduzido, a ordem do erro de discretização das

soluções numéricas tende à ordem assintótica dos erros de truncamento. A ordem efetiva pode ser calculada por:

$$p_E = \frac{\log \left[\frac{E(\phi_2)}{E(\phi_1)} \right]}{\log(q)} \quad (3.2)$$

onde ϕ_1 e ϕ_2 são as soluções numéricas obtidas nas malhas fina e grossa, respectivamente e q é a razão de refino de malha, dada por: $q = h_2/h_1$.

A ordem aparente permite verificar na prática se, à medida que o tamanho do volume de controle é reduzido, a ordem da incerteza das soluções numéricas tende à ordem assintótica dos erros verdadeiros. Para a obtenção da ordem aparente não é necessário conhecer a solução analítica, conforme pode ser observado na Eq. (3.3):

$$p_U = \frac{\log \left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2} \right)}{\log(q)} \quad (3.3)$$

onde ϕ_1 , ϕ_2 e ϕ_3 são as soluções numéricas obtidas para as malhas consideradas fina, grossa e super-grossa, respectivamente, e a razão de refino é dada por: $q = h_2/h_1 = h_3/h_2$.

Neste capítulo são apresentadas as ordens efetiva e aparente, bem como o erro numérico para as seguintes variáveis de interesse: temperatura no ponto central, temperatura média (obtida com a regra do trapézio em malhas quadrangulares e com a regra do retângulo em malhas triangulares), normas l_∞ e \bar{l}_1 do erro numérico.

A equação de Laplace é resolvida considerando condições de contorno do tipo Dirichlet, já descritas no Capítulo 2. Os resultados obtidos são para malhas variando de 5x5 até 4097x4097 incógnitas, incluindo os pontos que estão nos contornos. Para malhas quadrangulares, na resolução do sistema linear de equações, são considerados apenas os pontos internos do domínio de cálculo. As informações que estão nos contornos (condições de contorno), são repassadas para o termo fonte dos pontos internos, já que as temperaturas que estão nos contornos são conhecidas. Com isso, os sistemas lineares resolvidos variam de 3x3 até 4095x4095 incógnitas.

Para a análise das ordens de erro e do erro numérico, o processo iterativo é levado até que se atinja o erro de máquina, em todas as simulações. Este procedimento é adotado para verificar a oscilação da norma l_2 do resíduo em torno do erro de máquina. A partir desta oscilação podem ser determinadas as tolerâncias aplicadas na resolução dos problemas para os próximos capítulos.

As ordens de erro e o erro numérico são obtidos a partir do programa Richardson_3p1, de autoria do orientador desta tese. Este programa calcula as ordens efetiva e aparente, bem como os erros numéricos, a partir das soluções que entram no programa via bloco de notas. Para cada um dos tamanhos de problema resolvidos, tem-se uma saída de dados, onde estão gravados os valores obtidos para as variáveis de interesse.

3.1 Equação de Laplace – Malhas quadrangulares

A equação de Laplace é resolvida considerando um domínio de cálculo quadrado, unitário, com o método de diferenças finitas aplicado a malhas quadrangulares. Esta equação é resolvida com o emprego das duas condições de contorno, já descritas anteriormente. São calculadas as ordens efetiva e aparente e os erros numéricos para as variáveis de interesse detalhadas anteriormente: temperatura no ponto médio, temperatura média, normas l_∞ e \bar{l}_1 , ambas do erro. Os resultados podem ser observados nas Figs. 3.2 a 3.5. Nota-se que, para o problema linear, as ordens efetiva e aparente apresentam um comportamento diferente do apresentado pelo problema senoidal. Para este último problema, tanto a ordem efetiva quanto a ordem aparente obtidas para todas as variáveis de interesse, se aproximam de 2 com o refino da malha. Na estimativa *a priori*, a ordem assintótica (teórica) também é 2.

Para o problema linear, tanto a ordem efetiva como a aparente não tem seus valores tendendo para a ordem teórica. Em alguns casos, as ordens não podem nem ser calculadas, devido ao argumento do logaritmo, que pode ser observado nas Eqs. (3.2) e (3.3), ter sinal negativo. Este comportamento para o problema linear é esperado, pois não se tem a presença de erros de discretização, o que afeta o cálculo das ordens para o erro numérico. Desta forma, os resultados apresentados são os esperados para os dois problemas analisados: para o problema linear, onde não se tem a presença de erros de discretização, o cálculo para as ordens não faz sentido, e para o problema senoidal, as ordens aparente e efetiva, se aproximam da ordem assintótica com o refino da malha.

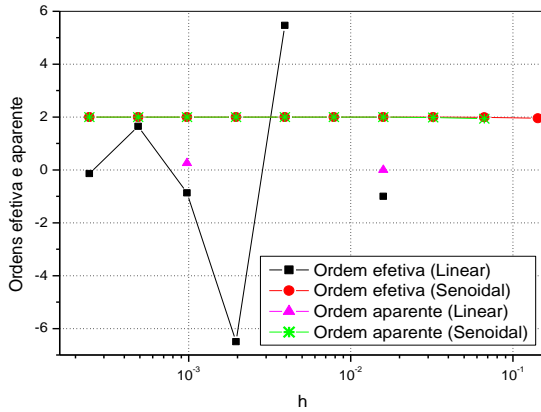


Figura 3.2: Ordens efetiva e aparente para temperatura no ponto médio para a equação de Laplace em malhas quadrangulares

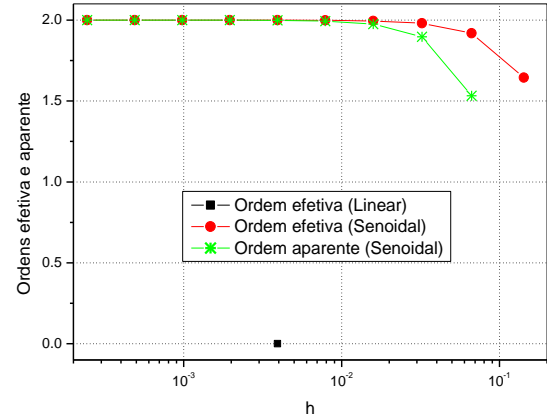


Figura 3.3: Ordens efetiva e aparente para temperatura média para a equação de Laplace em malhas quadrangulares

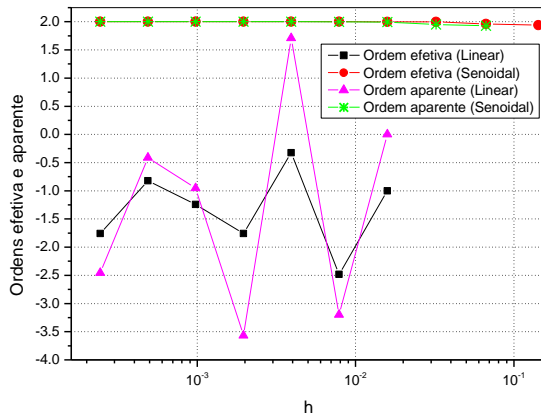


Figura 3.4: Ordens efetiva e aparente para a norma l_{∞} do erro para a equação de Laplace em malhas quadrangulares

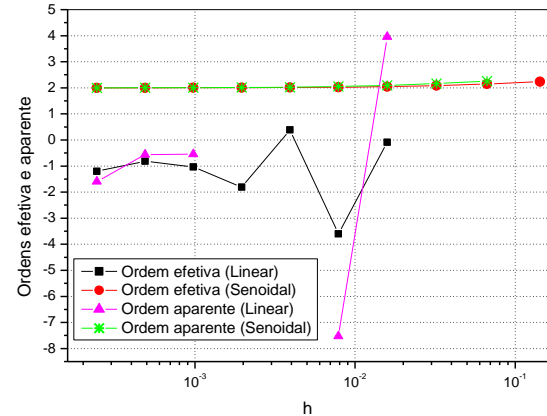


Figura 3.5: Ordens efetiva e aparente para a norma l_1 do erro para a equação de Laplace em malhas quadrangulares

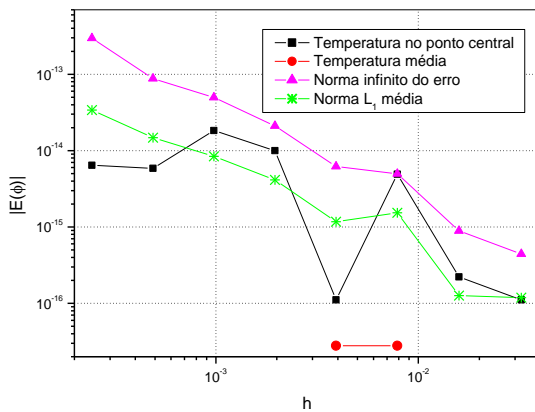


Figura 3.6: Erros numéricos para o problema linear em malhas quadrangulares

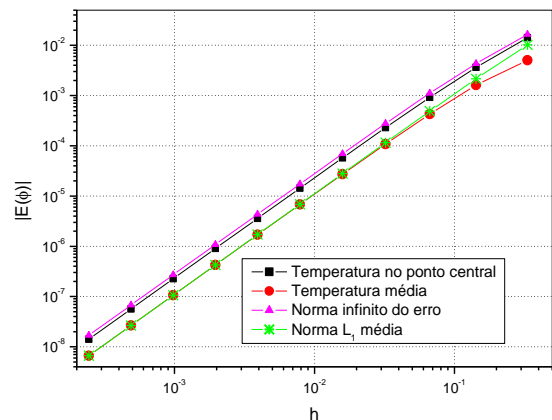


Figura 3.7: Erros numéricos para o problema senoidal em malhas quadrangulares

Nas Figs. 3.6 e 3.7 são apresentados os erros numéricos obtidos pela Eq. (3.1), para cada uma das variáveis de interesse, considerando os dois problemas resolvidos a partir da equação de Laplace. Na Fig. 3.6 podem ser observados os erros numéricos para todas as variáveis de interesse considerando o problema linear. Observa-se que o comportamento do erro de discretização é o esperado, pois quanto mais refinada a malha, maior é o erro. Este comportamento pode ser atribuído à influência do erro de arredondamento. Na Fig. 3.7, também pode ser observado o comportamento dos erros numéricos para todas as variáveis de interesse para o problema senoidal. Nota-se que o erro numérico está diminuindo com o refino da malha, comportamento esperado para este caso.

Na Fig. 3.8 pode ser observado o comportamento da norma l_∞ do erro conforme é variado o número de incógnitas, para os dois problemas resolvidos a partir da equação de Laplace. A norma l_∞ do erro apresenta o comportamento esperado para os dois casos em que é resolvida a equação de Laplace. Para o problema linear, seu valor aumenta com o crescimento do número de incógnitas, devido à influência do erro de arredondamento. Para o problema senoidal, o valor da norma l_∞ do erro está diminuindo.

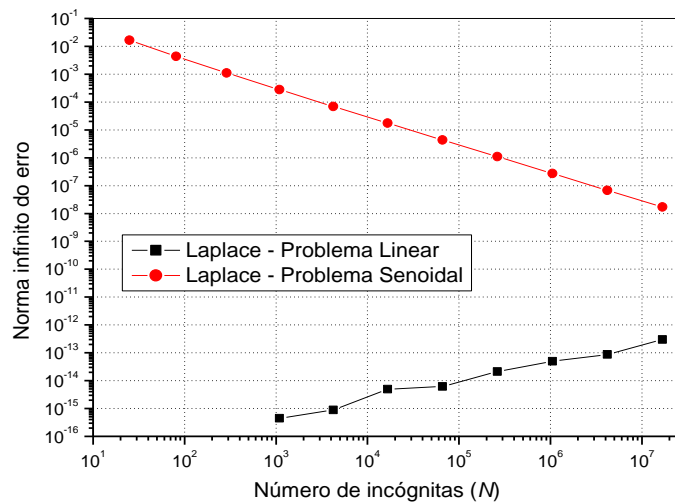


Figura 3.8: Comportamento da norma l_∞ do erro para a equação de Laplace em malhas quadrangulares

Durante a realização dos testes de coerência, a equação de Laplace também é resolvida com a aplicação do método *singlegrid* (método de malha única, SG) (FEDORENKO, 1964). São resolvidos problemas com até 513x513 incógnitas, devido ao tempo de CPU ser muito grande para problemas com mais incógnitas. Para o *singlegrid* também é calculada a norma l_∞ do erro, sendo que seu valor pode ser comparado ao obtido para o AMG. Para esta comparação são tomados os valores para o mesmo tamanho de problema, ou seja, com

513x513 incógnitas. Nesta comparação pode ser incluída a norma obtida para o método *multigrid* geométrico (GMG), que pode ser observada na Tab. 3.1:

Tabela 3.1: Comparativo da norma l_∞ do erro para a equação de Laplace em malhas quadrangulares

Problema	AMG	GMG	SG
Linear	2,10387263166E-14	1,11022302462E-15	2,39808173319E-14
Senoidal	1,08817138034E-06	1,08817137651E-06	1,08817070227E-06

Conforme observado na Tab. 3.1, para o problema linear, a norma l_∞ do erro obtida para cada uma das formas de se resolver o problema tem valores diferentes para AMG, GMG e SG. Esta diferença pode ser atribuída à influência do erro de arredondamento. Para o problema senoidal, observa-se que a norma l_∞ do erro é praticamente a mesma para o AMG, GMG e SG, conforme esperado. Isto quer dizer que, não importa o método empregado para a resolução do problema, a ordem do erro será a mesma para qualquer um deles.

Os testes computacionais realizados para se obter a tolerância a ser empregada para as próximas análises estão descritos a seguir. Para o problema linear, os testes são realizados para todos os tamanhos de problema, levando o processo iterativo até o erro de máquina ser atingido. Estes testes têm o objetivo de verificar quando o resíduo começa a oscilar em torno do erro de máquina. Com esta metodologia, verificou-se que, para a malha com mais incógnitas, a norma l_2 do resíduo, está em torno de 10^{-15} . Para o problema senoidal, o comportamento da norma l_2 do resíduo é parecido ao descrito acima, com a norma na malha com mais incógnitas oscilando em torno de 10^{-15} . Desta forma, a tolerância a ser empregada para a equação de Laplace é de $Tol = 10^{-13}$, pois com este valor, garante-se que a norma l_2 do resíduo não está oscilando em torno do erro de máquina.

Durante a execução destes testes computacionais também é monitorada a quantidade de memória computacional empregada na resolução dos problemas. O objetivo deste monitoramento é a determinação do maior problema (maior número de incógnitas) que pode ser resolvido sem o uso de memória virtual, ou seja, com o uso da memória física da máquina. Para a obtenção da quantidade de memória que estava sendo empregada para a resolução de cada um dos problemas é utilizado o Gerenciador de Tarefas do Windows, onde o uso de memória pode ser observado em Desempenho. Desta forma, pode-se concluir que, para a malha quadrangular, o maior problema a ser resolvido possui 4.097x4.097 incógnitas.

De acordo com Ruge e Stüben (1986), Briggs et al. (2000) e Trottenberg et al. (2001), não é vantajoso aplicar AMG a malhas estruturadas onde o GMG pode ser utilizado com

eficácia. Para estes casos, o GMG vai ser sempre mais rápido. Já o SG é sempre mais lento que o GMG e que o AMG. Estas afirmações são comprovadas neste trabalho, onde pode ser observado o comportamento dos tempos de CPU com o aumento do número de incógnitas para os três métodos acima citados.

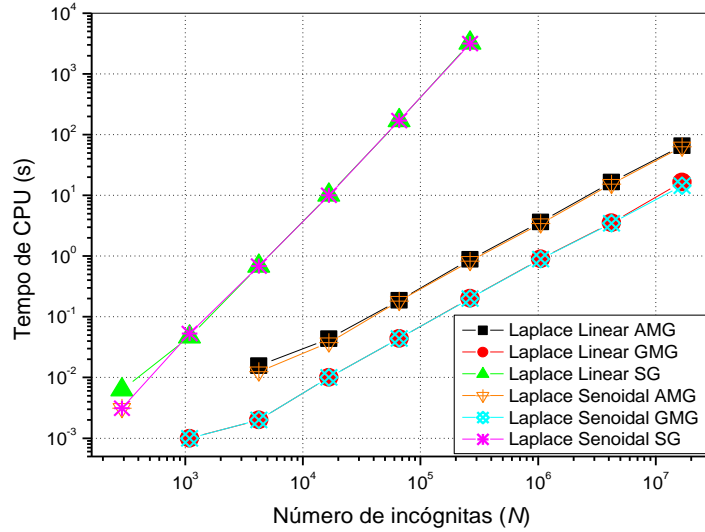


Figura 3.9: Tempo de CPU(s) versus número de incógnitas para a equação de Laplace em malhas quadrangulares

A Fig. 3.9 traz a comparação entre os tempos de CPU para a equação de Laplace resolvida com AMG, GMG e SG, tanto para o problema linear, quanto para o problema senoidal. Nesta mesma figura, pode-se observar que o tempo de CPU para o AMG apresenta o mesmo comportamento que o GMG, ou seja, ambos apresentam um crescimento próximo do linear e são muito menores que os valores apresentados para o *singlegrid*. Nesta mesma figura, pode-se observar ainda que o tempo de CPU para o GMG apresenta um valor menor para todos os tamanhos de problema, quando comparados ao AMG e ao SG, respectivamente.

O comportamento para AMG, GMG e SG apresentado na Fig. 3.9 é o esperado, conforme encontrado em Watanabe et al. (2005). Além disso, este é o resultado esperado pois para se resolver um problema com AMG, durante a fase *setup*, é necessária a criação de vários vetores (para armazenar as informações das malhas auxiliares), o que torna o procedimento mais lento e mais custoso.

A respeito da inclinação das curvas, é possível fazer um estudo com o ajuste de curvas pelo método dos mínimos quadrados. Em particular, o ajuste é feito por uma curva geométrica, conforme Eq. (3.4) abaixo:

$$t = cN^p \quad (3.4)$$

onde os parâmetros c e p são determinados pelo ajuste, N é o número de incógnitas e t é o tempo de CPU. Para os problemas aqui abordados foram tomados 4 pontos para que fosse feito o ajuste geométrico. Os valores correspondentes ao expoente p e ao coeficiente c podem ser observados na Tab. 3.2.

Tabela 3.2: Valores de c e p para AMG, GMG e SG para a equação de Laplace em malhas quadrangulares

Problema	Linear	Senoidal
AMG (c)	1,46565E-06	2,02806E-06
GMG (c)	2,63688E-07	2,98855E-07
SG (c)	2,12466E-08	9,68345E-09
AMG (p)	1,06361	1,05382
GMG (p)	1,08071	1,06874
SG (p)	1,92407	1,91452

Os valores para c e p da Tab. 3.2 ilustram o comportamento das curvas apresentadas na Fig. 3.9. Observa-se que para o AMG, os valores para c e para p estão próximos entre si, tanto para o problema linear quanto para o problema senoidal. Comportamento parecido é observado para o GMG e para o SG. Estes resultados concordam com os apresentados na Fig. 3.9, onde as curvas para o problema linear e senoidal, quando observados AMG, GMG e SG separadamente, estão muito próximas. Nota-se ainda que a inclinação para as curvas que representam o AMG e o GMG é muito próxima, visto que os expoentes p para estas curvas refletem este comportamento, estando próximos entre si. O afastamento entre elas é dado pelo coeficiente c , que apresenta valores diferentes para o AMG e o GMG. Os valores obtidos para o SG também refletem os resultados apresentados na Fig. 3.9. O valor para p é bem maior que os obtidos para o AMG e GMG, além do valor para c também apresentar uma grande diferença quando comparado aos valores obtidos para o AMG e GMG.

Com base nos resultados apresentados, pode-se concluir que o código computacional que resolve a equação de Laplace, quando são considerados o problema linear e o problema senoidal, em malhas quadrangulares, está coerente. As ordens do erro (ordem efetiva e aparente), calculadas para as variáveis de interesse (temperatura no ponto médio, temperatura média, norma l_∞ e \bar{l}_1 do erro numérico), apresentam o comportamento esperado. As ordens encontradas se aproximam de seus valores teóricos para o problema senoidal e são afetados pelo erro de arredondamento para o problema linear, sendo este último por não apresentar erros de discretização. Os erros numéricos calculados para cada uma das variáveis de interesse também apresentam o resultado esperado: diminuem conforme o refinamento da

malha, para o problema senoidal, e sendo afetados pelo erro de arredondamento para o problema linear. O mesmo resultado é verificado para a norma l_∞ do erro.

Quando são comparados os resultados obtidos para AMG, GMG e SG, para a norma l_∞ do erro, obtiveram-se os resultados esperados, ou seja, para a solução do problema senoidal, os valores para a norma l_∞ do erro são praticamente iguais, já para o problema linear, os valores são influenciados pelo erro de arredondamento. Quanto à comparação dos tempos de CPU, também se obteve o resultado esperado. O GMG é mais rápido que o AMG e o SG. Além disso, as curvas para o AMG e o GMG possuem praticamente a mesma inclinação, que é diferente para o SG.

3.2 Equação de Laplace – Malhas triangulares

A equação de Laplace também é discretizada com o uso de malhas triangulares, em um domínio de cálculo quadrado. São considerados novamente os dois problemas resolvidos a partir da equação de Laplace (problema linear e senoidal). Os mesmos testes computacionais feitos para se chegar às conclusões listadas em malhas quadrangulares são repetidos para malhas triangulares.

As ordens efetiva e aparente são calculadas para as mesmas variáveis de interesse já estudadas em malhas quadrangulares (temperatura no ponto médio, temperatura média, norma l_∞ e \bar{l}_1 do erro numérico). O comportamento para cada uma delas pode ser visualizado nas Figs. 3.10 a 3.13.

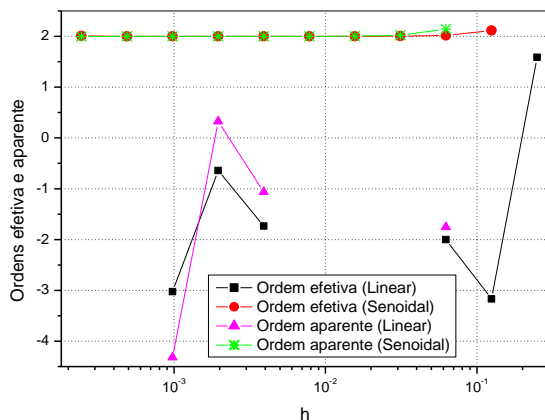


Figura 3.10: Ordens efetiva e aparente para a temperatura no ponto médio para a equação de Laplace em malhas triangulares

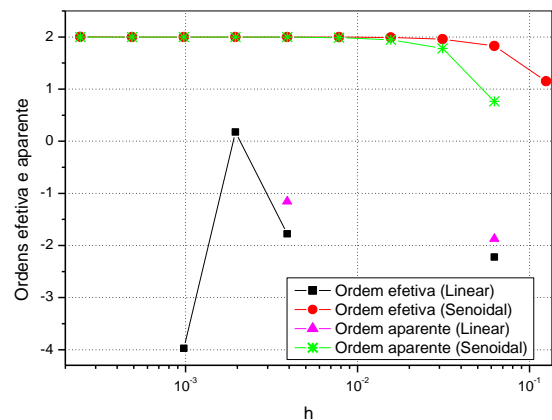


Figura 3.11: Ordens efetiva e aparente para a temperatura média para a equação de Laplace em malhas triangulares

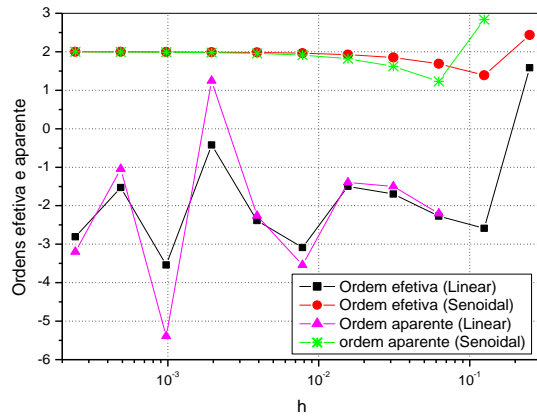


Figura 3.12: Ordens efetiva e aparente para a norma l_∞ do erro para a equação de Laplace em malhas triangulares

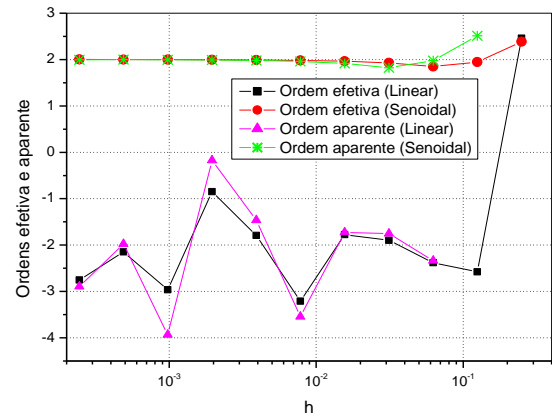


Figura 3.13: Ordens efetiva e aparente para a norma \bar{l}_1 do erro para a equação de Laplace em malhas triangulares

Em todos os casos observa-se o comportamento esperado para cada um dos problemas abordados. Para o problema linear, não faz sentido calcular as ordens do erro, devido à inexistência do erro de discretização. Os resultados apresentados para o problema senoidal também são coerentes, visto que as ordens efetiva e aparente obtidas estão tendendo ao valor teórico (ordem assintótica), que é igual a 2.

Nas Figs. 3.14 e 3.15 pode-se observar o comportamento do erro numérico com o refino da malha para cada uma das variáveis de interesse. Nota-se que este comportamento é parecido ao obtido para malhas quadrangulares. Para o problema linear (Fig. 3.14), quanto mais refinada a malha, maior é o erro, devido à influência do erro de arredondamento. Para o problema senoidal (Fig. 3.15), o erro numérico diminui conforme a malha vai sendo refinada.

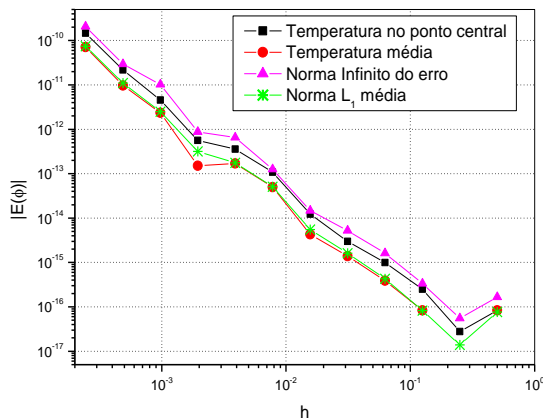


Figura 3.14: Erros numéricos para o problema linear em malhas triangulares

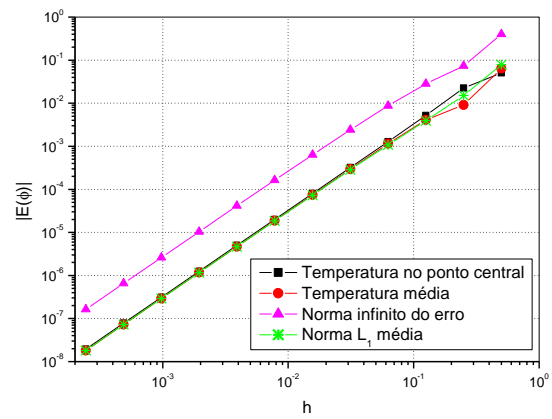


Figura 3.15: Erros numéricos para o problema senoidal em malhas triangulares

Na Fig. 3.16, o comportamento da norma l_∞ do erro conforme é variado o número de incógnitas pode ser observado. Este comportamento é o esperado, pois para o problema linear,

a norma l_∞ do erro aumenta com o número de incógnitas, devido à influência de erros de arredondamento. Para o problema senoidal, seu valor diminui com o crescimento do número de incógnitas.

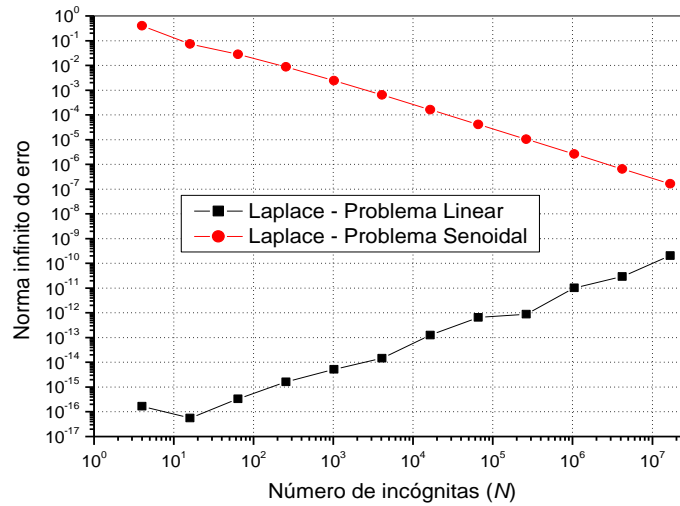


Figura 3.16: Comportamento da norma l_∞ do erro para a equação de Laplace em malhas triangulares

Para malhas triangulares, a equação de Laplace também é resolvida com o método *singlegrid*, sendo que a norma l_∞ do erro pode ser comparada com a obtida para o AMG, na malha com $2^{18} = 262.144$ incógnitas, que é o maior problema resolvido com o *singlegrid*.

Tabela 3.3: Comparativo da norma l_∞ do erro para a equação de Laplace em malhas triangulares

Problema	AMG	SG
Linear	8,76521077942E-13	2,39097630583E-12
Senoidal	1,03939171850E-05	1,03939172044E-05

Na Tab. 3.3 podem ser observados os valores obtidos para a norma l_∞ do erro para as duas abordagens da equação de Laplace em malhas triangulares. Para o problema linear, o resultado para a norma l_∞ do erro é o esperado, visto que seus valores estão sofrendo a influência do erro de arredondamento. Já para o problema senoidal, os valores obtidos para a norma l_∞ do erro são praticamente iguais, conforme esperado. Este comparativo está de acordo com as análises que foram feitas para malhas quadrangulares, descritas na seção anterior.

Para malhas triangulares também são realizados testes computacionais cujo objetivo é a obtenção da tolerância a ser adotada para o critério de parada. Para o problema linear, a

norma l_2 do resíduo, está em torno de 10^{-15} . Para o problema senoidal, este valor está oscilando em 10^{-16} . Desta forma, o valor a ser adotado como tolerância para estes casos deve ser $Tol = 10^{-13}$, pois desta forma garante-se que o resíduo não irá oscilar em torno do erro de máquina.

Uma comparação entre os tempos de CPU para o AMG e o SG também é feita para malhas triangulares, conforme pode ser observado na Fig. 3.17. As curvas dos tempos de CPU para cada um dos problemas resolvidos com AMG apresentam o comportamento esperado, ou seja, estão bem próximas e apresentam praticamente a mesma inclinação. Para as curvas que representam o SG o comportamento também é o esperado. As curvas que ilustram o AMG e o SG estão afastadas, conforme o esperado.

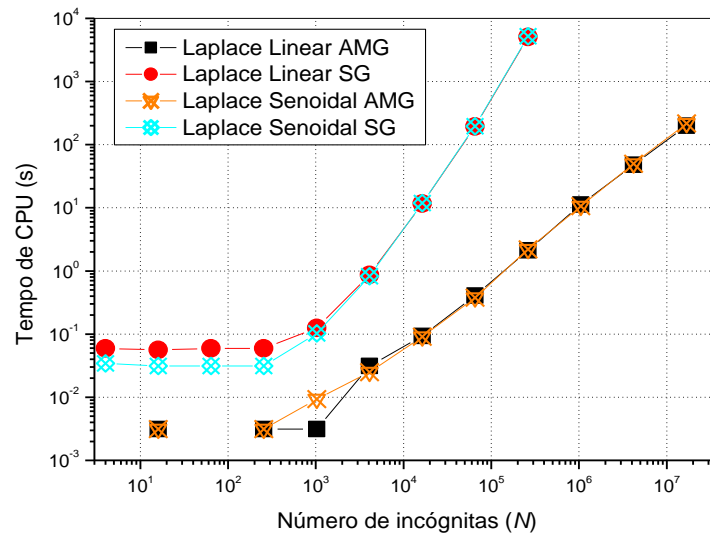


Figura 3.17: Tempo de CPU(s) versus número de incógnitas para a equação de Laplace em malhas triangulares

Para complementar os resultados apresentados acima, são calculados os valores para c e p da Eq. (3.4). Estes valores podem ser observados na Tab. 3.4.

Tabela 3.4: Valores de c e p para AMG, GMG e SG para a equação de Laplace em malhas triangulares

Problema	Linear	Senoidal
AMG (c)	2,95447E-06	2,02806E-06
SG (c)	2,26339E-08	8,96788E-08
AMG (p)	1,07737	1,07357
SG (p)	1,87915	1,88590

Os valores para c e p da Tab. 3.4 ilustram o comportamento das curvas apresentadas na Fig. 3.17. Observa-se que para o AMG, os valores para c e para p estão próximos entre si,

tanto para o problema linear quanto para o problema senoidal. Comportamento parecido é observado para o SG. Estes resultados concordam com os apresentados na Fig. 3.17, onde as curvas para os problemas abordados, quando observados AMG e SG separadamente, estão muito próximas. O afastamento entre elas é dado pelo coeficiente c , que apresenta valores diferentes para o AMG e o SG.

Todos os resultados obtidos para malhas triangulares concordam com os obtidos para malhas quadrangulares. Com base nisto, pode-se concluir que o código computacional empregado na resolução da equação de Laplace com malhas triangulares está coerente. Na sequência, a mesma análise é feita para a equação de Poisson discretizada em malhas quadrangulares.

3.3 Equação de Poisson – Malhas quadrangulares

A equação de Poisson é discretizada em um domínio de cálculo quadrado, unitário, com o uso do método de diferenças finitas, aplicado a malhas quadrangulares, cujas condições de contorno são de Dirichlet. Para esta equação são calculadas as ordens efetiva e aparente para todas as variáveis de interesse já listadas anteriormente (temperatura no ponto médio, temperatura média, normas l_∞ e \bar{l}_1 do erro), bem como os erros numéricos para cada uma delas. Os resultados obtidos podem ser observados nas Figs. 3.18 a 3.21.

Nas Figs. 3.18 e 3.19 podem ser observadas as ordens efetiva e aparente para cada variável de interesse. O comportamento destas ordens é o esperado, pois seus valores se aproximam do valor teórico (ordem assintótica é 2) com o refinamento da malha.

Na Fig. 3.20 pode ser observado o comportamento do erro numérico. Seu valor está caindo com o refino da malha. Estes resultados concordam com os obtidos para a equação de Laplace, discretizada tanto com o uso de malhas quadrangulares quanto de malhas triangulares. Para cada um dos tamanhos de problema resolvidos, foi tomado o valor da norma l_∞ do erro sendo que estes valores estão ilustrados na Fig. 3.21.

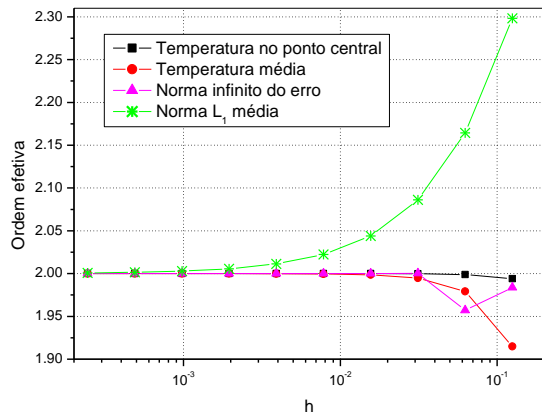


Figura 3.18: Ordem efetiva – equação de Poisson em malhas quadrangulares

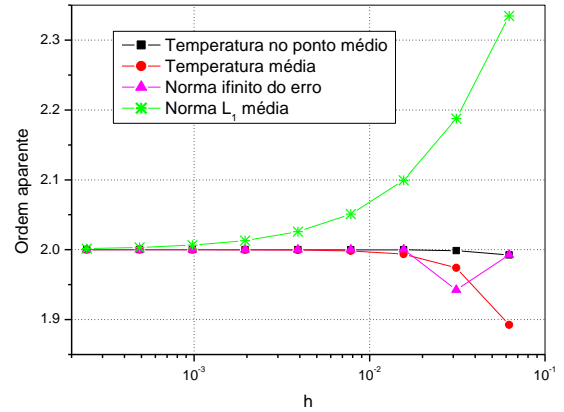


Figura 3.19: Ordem aparente – equação de Poisson em malhas quadrangulares

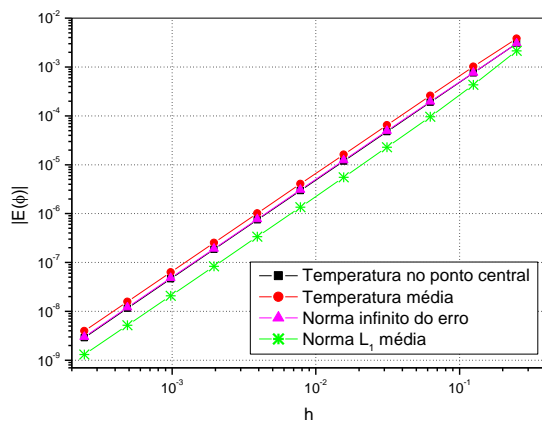


Figura 3.20: Erros numéricos – equação de Poisson em malhas quadrangulares

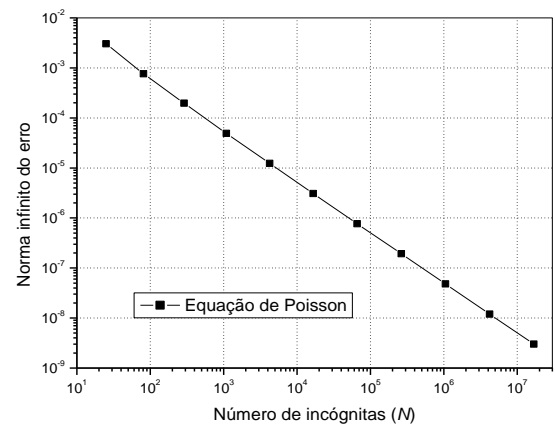


Figura 3.21: Norma l_{∞} do erro para a equação de Poisson em malhas quadrangulares

Na Fig. 3.21, pode-se observar que o comportamento obtido para a norma l_{∞} do erro tem o comportamento esperado, ou seja, seu valor diminui conforme aumenta o número de incógnitas. A equação de Poisson em malhas quadrangulares também foi resolvida com o método *singlegrid*, sendo que a norma l_{∞} do erro também pode ser comparada com a obtida para o AMG, na malha com 513x513 incógnitas, que é o maior problema resolvido com o método *singlegrid*. Este valor também pode ser comparado ao obtido para o GMG. Esta comparação pode ser observada na Tab. 3.5.

Tabela 3.5: Comparativo da norma l_{∞} do erro para a equação de Poisson

Problema	AMG	GMG	SG
Poisson	1,92072535661E-07	1,92072544640E-07	1,92072607312E-07

Conforme Tab. 3.5, para a equação de Poisson, os valores encontrados para a norma l_∞ do erro para as três formas de se resolver o problema estão muito próximos, conforme o esperado. Este resultado concorda com os obtidos o problema senoidal, resolvido a partir da equação de Laplace, em ambas as malhas, quadrangulares e triangulares.

Para a equação de Poisson também são realizados testes computacionais cujo objetivo é a obtenção da tolerância a ser adotada para o critério de parada. Para a malha com mais incógnitas, a norma l_2 do resíduo, está em torno de 10^{-10} . Desta forma, empregando o mesmo raciocínio descrito acima para a equação de Laplace, o valor a ser adotado como tolerância para este caso deve ser $Tol = 10^{-8}$, pois com este valor garante-se que o resíduo não irá oscilar em torno do erro de máquina.

São comparados os tempos de CPU para AMG, GMG e SG, sendo que os resultados obtidos são os esperados. Estes resultados coincidem com os obtidos para a equação de Laplace discretizada tanto em malhas quadrangulares (com AMG e GMG) quanto triangulares (com AMG). Esta comparação pode ser observada na Fig. 3.22.

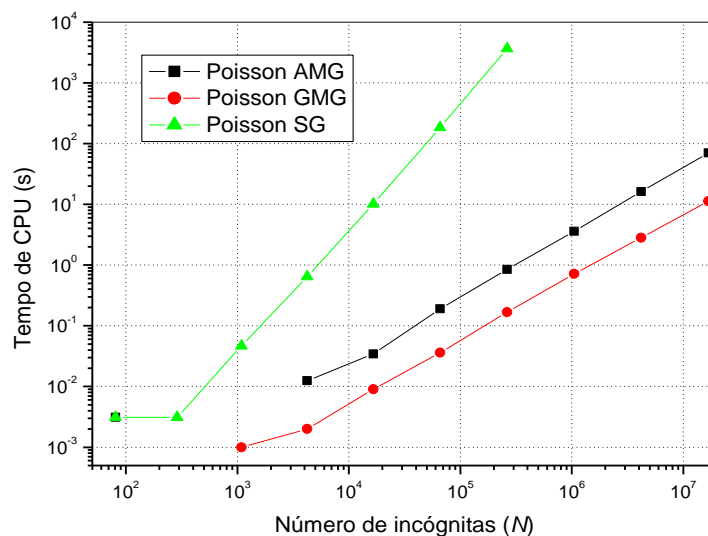


Figura 3.22: Tempo de CPU(s) versus número de incógnitas para a equação de Poisson em malhas quadrangulares

Os resultados apresentados na Fig. 3.22 estão de acordo com o esperado para os três casos analisados: AMG, GMG e SG. Nesta figura nota-se que as curvas obtidas para o AMG e o GMG estão um pouco afastadas, mas apresentam praticamente a mesma inclinação. A curva que representa o SG além de estar mais afastada das outras duas, ainda apresenta uma inclinação maior. Observa-se ainda que o GMG é mais rápido que o AMG e que estes dois,

por sua vez, são mais rápidos que o SG. Para complementar os resultados apresentados, são calculados os valores para c e p da Eq. (3.4). Estes valores podem ser observados na Tab. 3.6.

Tabela 3.6: Valores de c e p para AMG, GMG e SG para a equação de Poisson em malhas quadrangulares

Problema	Poisson
AMG (c)	1,37148E-06
GMG (c)	3,45772E-07
SG (c)	2,78535E-08
AMG (p)	1,12084
GMG (p)	1,04387
SG (p)	2,04112

Os valores para c e p da Tab. 3.6 ilustram o comportamento das curvas apresentadas na Fig. 3.22. Nota-se ainda que a inclinação para as curvas que representam o AMG e o GMG é muito próxima, visto que os expoentes p para estas curvas refletem este comportamento, estando próximos entre si. O afastamento entre elas é dado pelo coeficiente c , que apresenta valores diferentes para o AMG e o GMG. Os valores obtidos para o SG também refletem os resultados apresentados na Fig. 3.22. O valor para p é bem maior que os obtidos para o AMG e GMG, além do valor para c também apresentar uma grande diferença quando comparado aos valores obtidos para o AMG e GMG.

Com base nos resultados obtidos para a análise de erros numéricos chega-se à conclusão de que o código computacional está correto, já que os resultados para os testes descritos mostram-se coerentes. Além disso, foi encontrada a tolerância a ser adotada para a execução dos testes computacionais. Para a equação de Laplace, tanto em malhas quadrangulares quanto triangulares, a tolerância a ser empregada deve ser $Tol = 10^{-13}$. Para a equação de Poisson, este valor deve ser $Tol = 10^{-8}$. Como um dos objetivos deste trabalho é a comparação dos resultados obtidos para todas as equações, fica estabelecido que a tolerância a ser adotada é $Tol = 10^{-8}$ para todos os problemas a serem resolvidos.

4 Resultados para malhas estruturadas quadrangulares

Neste capítulo são mostrados os resultados obtidos para a equação de Laplace e Poisson, ambas discretizadas em malhas estruturadas quadrangulares, com o uso do método de diferenças finitas. Estas equações são resolvidas com o uso dos métodos *multigrid* algébrico (AMG), *multigrid* geométrico (GMG) e *singlegrid* (SG). A avaliação dos parâmetros do método *multigrid* algébrico é feita com o objetivo de verificar sua influência no tempo de CPU para a resolução destes problemas. Os parâmetros avaliados são: número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). É avaliado também o efeito do tempo de CPU com o aumento do número de incógnitas (N). Com a otimização do algoritmo obteve-se uma redução no tempo de CPU e no número de iterações externas (ciclos V) para todos os problemas aqui estudados. O critério de parada adotado é a norma l_2 do resíduo, sendo que a tolerância adotada é de 10^{-8} , conforme determinado no Capítulo 3.

São resolvidas a equação de Poisson e os dois problemas que envolvem a equação de Laplace (problema linear e senoidal), sendo que para todos eles a solução analítica é conhecida, conforme descrito no capítulo 2. Para todas as equações descritas são feitos testes computacionais para que sejam encontrados valores ótimos para os parâmetros especificados acima, sendo que, para alguns deles, é possível fazer um comparativo entre os resultados obtidos para os métodos *multigrid* algébrico e geométrico.

Para a obtenção dos resultados são considerados três tamanhos de problemas: pequeno, médio e grande, com 257×257 , 1025×1025 e 4097×4097 incógnitas, respectivamente. Para os problemas considerados pequeno e médio, para os quais o tempo de CPU obtido para as simulações é inferior a 10 segundos, as simulações são repetidas 3 vezes, sendo que é tomada a média aritmética destes tempos. Para malhas quadrangulares, com o uso do AMG, não foram encontrados trabalhos que tragam resultados em malhas tão refinadas como as aqui apresentadas.

Para um melhor entendimento, este capítulo está dividido em quatro seções: equação de Laplace - Problema linear; equação de Laplace – Problema senoidal; equação de Poisson; e comparativo para malhas quadrangulares.

4.1 Equação de Laplace – Problema linear

Para esta equação são feitos testes para se encontrar o valor ótimo para: número de iterações internas (ν), número de malhas (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). Entende-se por valor ótimo como aquele que corresponde ao menor tempo de CPU encontrado para o parâmetro em questão, quando todos os outros são considerados constantes. Depois de definir as componentes algorítmicas ótimas, chega-se a um algoritmo otimizado. O algoritmo com as componentes indicadas pela literatura é o algoritmo padrão. Para estes dois algoritmos são comparados os tempos de CPU, o coeficiente c e o expoente p (que indica a inclinação das curvas do tempo de CPU *versus* número de incógnitas), da Eq. (3.4) e o número de iterações externas empregadas até que se atinja a tolerância desejada.

4.1.1 Número de iterações internas (ν)

Para encontrar o valor ótimo para o número de iterações internas (ν) são considerados todos os níveis de malha, ou seja, $L = L_{máximo}$. Para o fator de redução de malha é empregado $\theta = 0,25$, e para o fator de forte dependência na malha grossa, $\varepsilon = 0,35$, que são os valores considerados padrão pela literatura. Os valores para ν estão variando entre 1 e 20. Este intervalo é o escolhido, pois, para o GMG (Oliveira et al., 2008) o valor ótimo para ν está entre estes valores. Na Fig. 4.1 pode ser observada a influência do número de iterações internas no tempo de CPU para os tamanhos de problema especificados acima. Esta figura mostra também o valor ótimo para este parâmetro, que é o valor que resulta no menor tempo de CPU.

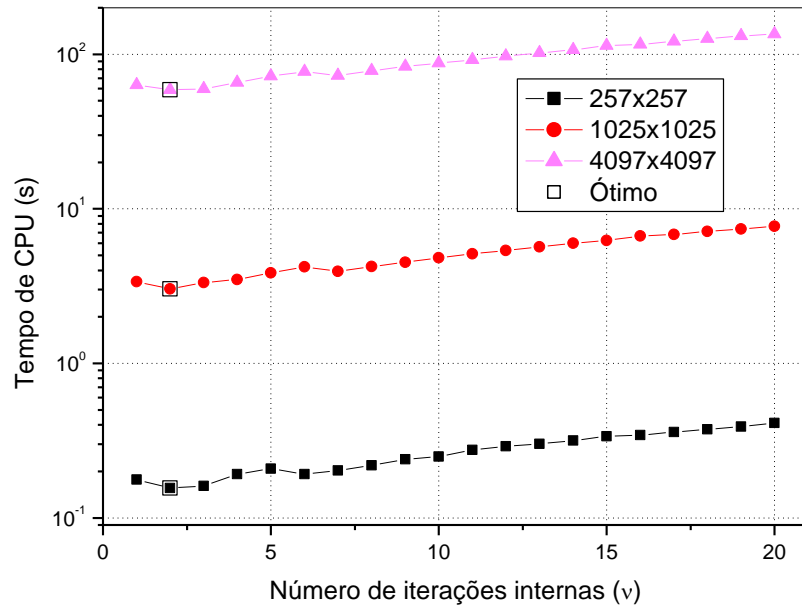


Figura 4.1: Tempo de CPU *versus* número de iterações internas para o problema linear em malhas quadrangulares

Conforme observado na Fig. 4.1, independente do tamanho de problema que está sendo resolvido, o número ótimo de iterações internas para o AMG é 2, ou seja, $\nu_{\text{ótimo}} = 2$. Observa-se que, geralmente, quanto maior o número de iterações internas, maior é o tempo de CPU gasto para se resolver o problema.

Em Ruge e Stüben (1986), Krechel e Stüben (1999), Briggs et al. (2000), Trottenberg et al. (2001), Iwamura et al. (2003) e Falgout (2006) é empregado $\nu = 1$ para a resolução de problemas com o AMG. Desta forma, o ganho em tempo de CPU, comparando o valor ótimo com o valor amplamente empregado pela literatura, $\nu = 1$, fica em 13,3%, 11,3% e 7,5%, para os problemas considerados pequeno, médio e grande, respectivamente.

Para o problema linear, observa-se que o valor de $\nu_{\text{ótimo}}$ coincide com o encontrado por Oliveira et al. (2008), para este mesmo problema, resolvido com GMG. Com base nas análises feitas acima, pode-se concluir que o valor a ser empregado nas próximas simulações é o número ótimo de iterações internas, ou seja, $\nu_{\text{ótimo}} = 2$.

4.1.2 Número de níveis (L)

Para o estudo da influência do número de níveis no tempo de CPU é utilizado o valor ótimo encontrado para o número de iterações internas. Desta forma, os parâmetros

empregados são: $\nu_{\text{ótimo}} = 2$, fator de redução de malha, $\theta = 0,25$ e fator de forte dependência na malha grossa, $\varepsilon = 0,35$.

Os valores empregados para o número de níveis são variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia com o tamanho do problema, ou seja, $L_{\text{máximo}}, L_{\text{máximo}} - 1, L_{\text{máximo}} - 2$, e assim sucessivamente. Para a determinação deste valor, é observado o crescimento do tempo de CPU, com a diminuição do número de níveis. O valor para L tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele em que a resolução do problema leva em torno de 3 horas. Deve-se observar que com este valor adotado já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar. Se fossem feitos testes com outros valores para L (menores que o mínimo aqui adotado), os tempos de CPU resultantes estariam entre 20 e 24 horas, valores que dificultariam a visualização gráfica deste parâmetro.

Na Fig. 4.2 pode-se observar a influência do número de níveis no tempo de CPU. Nota-se ainda a existência de um valor ótimo deste parâmetro para cada um dos tamanhos de problema analisados. Nesta figura nota-se também a tendência do comportamento para o número de níveis.

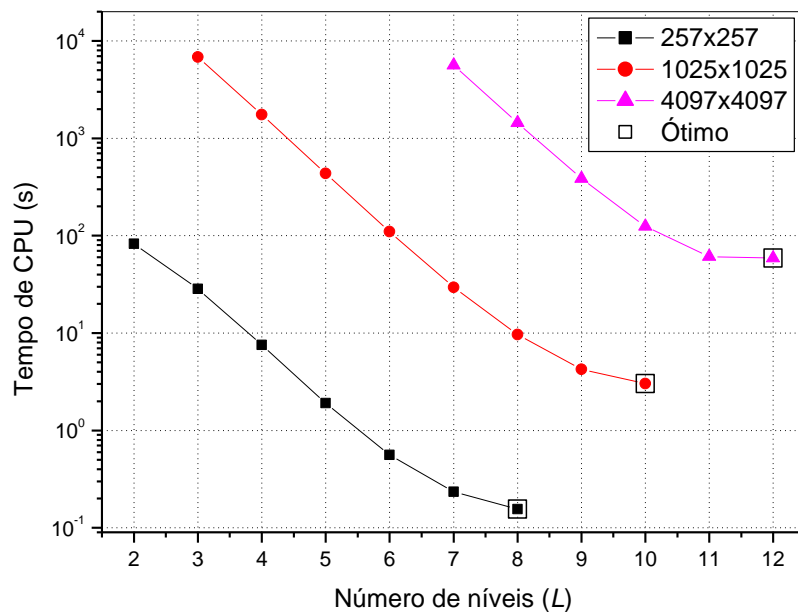


Figura 4.2: Tempo de CPU versus número de níveis para o problema linear em malhas quadrangulares

Os resultados observados na Fig. 4.2 para o número de níveis no problema de minimização do tempo de CPU são os seguintes: para $N=257 \times 257$, que tem $L_{\text{máximo}} = 8$, tem-

se $L_{\text{ótimo}} = 8$, para $N = 1025 \times 1025$, que tem $L_{\text{máximo}} = 10$, tem-se $L_{\text{ótimo}} = 10$ e para $N = 4097 \times 4097$ que tem $L_{\text{máximo}} = 12$ tem-se $L_{\text{ótimo}} = 12$. Desta análise pode-se concluir que existe um número de níveis que resulta no menor tempo de CPU. Este valor é o número máximo de níveis, que varia conforme o tamanho do problema. Desta forma, o valor a ser empregado nas próximas análises é $L_{\text{ótimo}} = L_{\text{máximo}}$. Conclui-se também que o tamanho de problema a ser resolvido não influencia no valor a ser empregado para o número de níveis. Observa-se ainda que, quanto menor o número de níveis de malha empregados na resolução, maior é o tempo de CPU para o AMG.

Estes resultados concordam com os obtidos para o GMG em Oliveira et al. (2008), onde é apresentado um estudo a respeito da influência deste parâmetro no tempo de CPU. Para o GMG, o número de níveis ótimo é o máximo. Durante a revisão bibliográfica, observou-se que alguns autores, como Ruge e Stüben (1986), Falgout (2006), Langer e Pusch (2006) não se preocuparam com o número de malhas a ser empregado. Nos trabalhos em que este dado está explícito, nota-se a utilização de todos os níveis de malha, como por exemplo em Krechel e Stüben (1999), Iwamura et al. (2003) e em Wu e Elman (2006). Não foram encontrados na literatura estudos a respeito do efeito do número de níveis de malha no tempo de CPU para o AMG.

4.1.3 Fator de redução de malha (θ)

Para o estudo deste parâmetro são considerados os valores ótimos encontrados para o número de iterações internas, $\nu = 2$, e para o número de malhas, $L = L_{\text{máximo}}$. Para o fator de forte dependência na malha grossa, é empregado o valor considerado padrão, $\varepsilon = 0,35$. O fator de redução de malha é uma constante que quantifica o quanto um ponto está fortemente conectado a outro. Dependendo desta relação, um ponto pode ser considerado da malha grossa ou não. Para malhas estruturadas quadrangulares o valor empregado para θ não influencia no número de níveis gerados na fase *setup*, ou seja, para todos os valores aqui testados para θ , o número de níveis empregado é sempre o mesmo para cada tamanho de problema.

Existe discordância com relação aos valores empregados pelos autores que resolvem problemas empregando AMG. A maioria dos trabalhos, como Ruge e Stüben (1986), Krechel e Stüben (1999), Trottenberg et al. (2001) e Iwamura et al. (2003) empregaram $\theta = 0,25$. Briggs et al. (2000) empregaram $\theta = 0,20$ e Falgout (2006) usou $\theta = 0,4$. Nenhum dos

trabalhos pesquisados justificou a escolha do valor para θ , apenas ressaltaram que $\theta = 0,25$ é um valor padrão. Também não foram encontrados trabalhos que tragam um estudo comparativo para diferentes valores de θ .

Por analogia ao GMG, observa-se que, para uma malha estruturada, $\theta = 0,25$ ou $\theta = 1/4$ corresponde à interpolação de quatro pontos (norte, sul, leste e oeste) para se obter o ponto da malha imediatamente mais fina, o que corresponde à razão de engrossamento $r = 2$ para o GMG. Desta forma, $\theta = 0,25$ quer dizer que um dado ponto está fortemente conectado a outros quatro. Para $\theta = 0,2$ (ou $\theta = 1/5$), são necessários 5 pontos para a interpolação, por exemplo. Os outros valores para o fator de redução de malha aqui testados têm correspondência com o GMG: $\theta = 0,11$ que corresponde à razão de engrossamento $r = 3$ e $\theta = 0,0625$, que corresponde à razão $r = 4$. Além destes valores, será testado ainda $\theta = 0,4$, que é empregado por Falgout (2006). Na Fig. 4.3 pode ser observado o comportamento do fator de redução de malha conforme a variação do número de incógnitas.

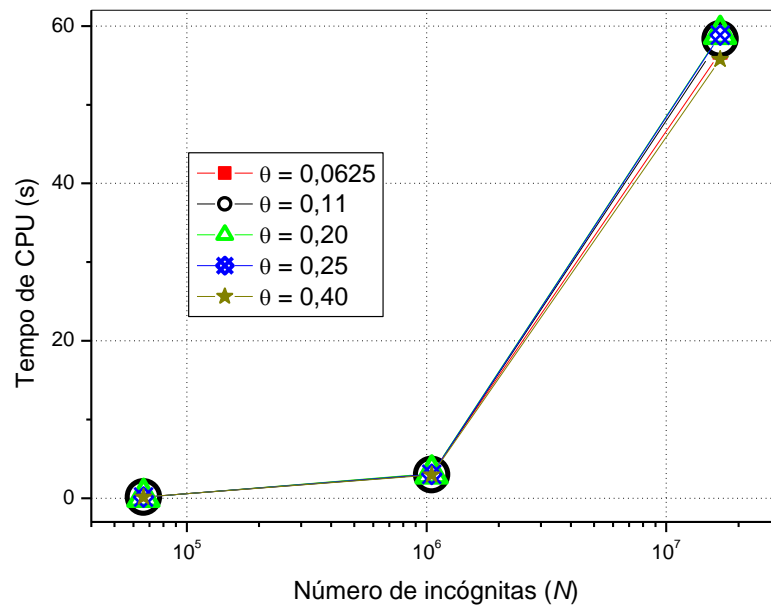


Figura 4.3: Influência do fator de redução de malha no tempo de CPU para o problema linear em malhas quadrangulares

Conforme observado na Fig. 4.3, a diferença entre os valores testados para o fator de redução de malha é muito pequena. São obtidos os tempos para os diferentes valores de θ , sendo que, para cada tamanho de problema tem-se um valor ótimo para este parâmetro. Desta forma, é calculada a diferença (em termos de porcentagem), entre o valor ótimo e cada um dos valores de θ . Estas porcentagens podem ser observadas na Tab. 4.1.

Tabela 4.1: Diferenças entre os tempos de CPU para θ considerando o problema linear em malhas quadrangulares

θ	257x257	1025x1025	4097x4097
0,0625	10%	3,8%	1,6%
0,11	3,3%	0,2%	4,6%
0,2	6,7%	4,2%	5,7%
0,25	-----	1,6%	5,5%
0,4	-----	-----	-----

Da análise da Tab. 4.1 pode-se concluir que o valor que apresenta o melhor desempenho dentre todos os testados, é $\theta = 0,4$. Para este valor de θ , o tempo de CPU é o menor, para os três tamanhos de problema analisados. Desta forma, para o problema linear, observa-se a existência de um valor ótimo para θ . Este valor é $\theta_{\text{ótimo}} = 0,4$. Desta tabela também se pode concluir que, o valor amplamente empregado na literatura ($\theta = 0,25$), não apresenta um comportamento satisfatório, pois as porcentagens que expressam a diferença entre o valor ótimo e seu valor, aumentam conforme o número de incógnitas é incrementado, apesar de apresentar o valor ótimo para a malha com menos incógnitas. Para o problema linear, um estudo complementar para este parâmetro é apresentado no Apêndice B.

4.1.4 Fator de forte dependência na malha grossa (ε)

Para a obtenção do valor ótimo para este parâmetro, são considerados os valores ótimos obtidos para: número de iterações internas $\nu = 2$, número de malhas, $L = L_{\text{máximo}}$ e fator de redução de malha, $\theta = 0,4$. O fator de forte dependência na malha grossa é uma constante que define a forte dependência no conjunto de pontos que estão na malha grossa. Se este valor for grande, quer dizer que os pontos em F com os quais o ponto i tem forte conexão negativa, necessitam ter uma forte dependência do ponto em C para serem incluídos na interpolação. Esta constante exerce influência na construção das malhas auxiliares para a aplicação do AMG.

Este valor não é divulgado em alguns trabalhos que tratam do AMG, como em Watanabe et al. (2005), Campos et al. (2006) e Langer e Pusch (2006). Em Ruge e Stüben (1986), Chang et al. (1996), Krechel e Stüben (1999), Briggs et al. (2000), Cleary et al. (2000) e Falgout (2006) é utilizado $\varepsilon = 0,35$. Este valor é o mais utilizado na literatura, é o valor

considerado padrão. Em Iwamura et al. (2003), são empregados valores diferentes conforme o nível de malha que está sendo construído, desta forma, tem-se $\varepsilon = 0,35$ para o número de níveis menor ou igual a 3 e, $\varepsilon = 0,45$ quando o número de níveis for maior que 3. Em Trottenberg et al. (2001) é utilizado $\varepsilon = 0,20$. No presente trabalho foram testados vários valores para ε . Na Fig. 4.4 pode ser observado o comportamento para os valores testados de ε , com o aumento do número de incógnitas.

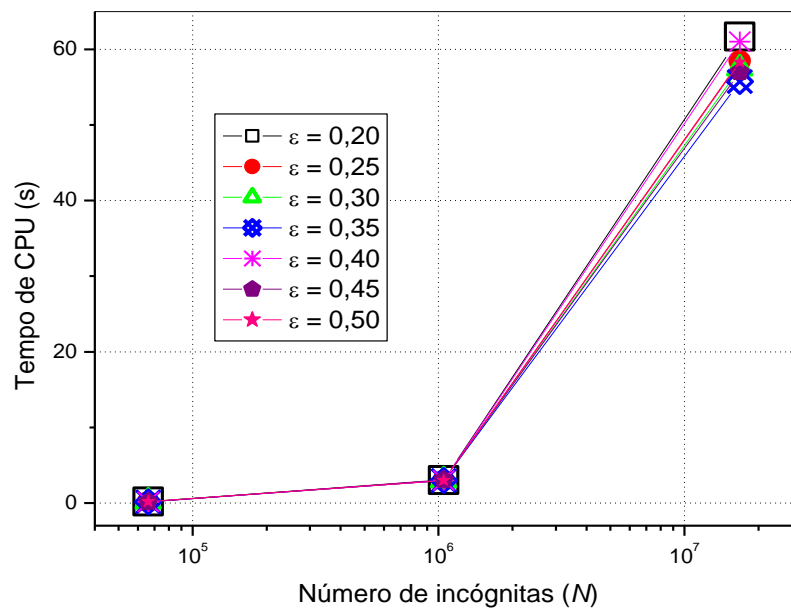


Figura 4.4: Efeito do fator de forte dependência na malha grossa no tempo de CPU para o problema linear em malhas quadrangulares

Pela Fig. 4.4 observa-se que, $\varepsilon = 0,35$ apresenta o menor tempo de CPU com o aumento do número de incógnitas. Este resultado pode ser confirmado através da Tab. 4.2, que traz as porcentagens para os diferentes valores usados para ε , para os três tamanhos de problema resolvidos.

Tabela 4.2: Diferenças entre os tempos de CPU para ε considerando o problema linear em malhas quadrangulares

ε	257x257	1025x1025	4097x4097
0,20	16,7%	1,6%	10,6%
0,25	-----	0,3%	4,8%
0,30	7,4%	-----	3,1%
0,35	-----	0,3%	-----
0,40	-----	-----	9,4%
0,45	3,3%	3,3%	2,2%
0,50	7,4%	0,3%	4,6%

Dentre todos os valores testados para ε , observa-se que o valor que apresenta as menores porcentagens é $\varepsilon = 0,35$, conforme observação da Fig. 4.4. Este valor é o ótimo para os problemas considerados pequeno e grande. Observa-se também pela Tab. 4.2 que os outros valores empregados para ε apresentam valores pequenos para as porcentagens (menos que 11% para a malha com mais incógnitas). Isto significa que este valor afeta pouco o desempenho do método, para o problema em estudo. O valor a ser empregado nas próximas simulações é aquele que apresenta o melhor desempenho médio, ou seja, $\varepsilon = 0,35$.

4.1.5 Número de incógnitas (N)

O problema linear é resolvido para diversos tamanhos de malha (de 5×5 até 4097×4097 incógnitas). Os resultados obtidos para o AMG são comparados aos obtidos para o GMG e para o SG. Esta comparação pode ser observada na Fig. 4.5. Desta forma, é feita a avaliação do efeito do número de incógnitas (N) sobre o tempo de CPU.

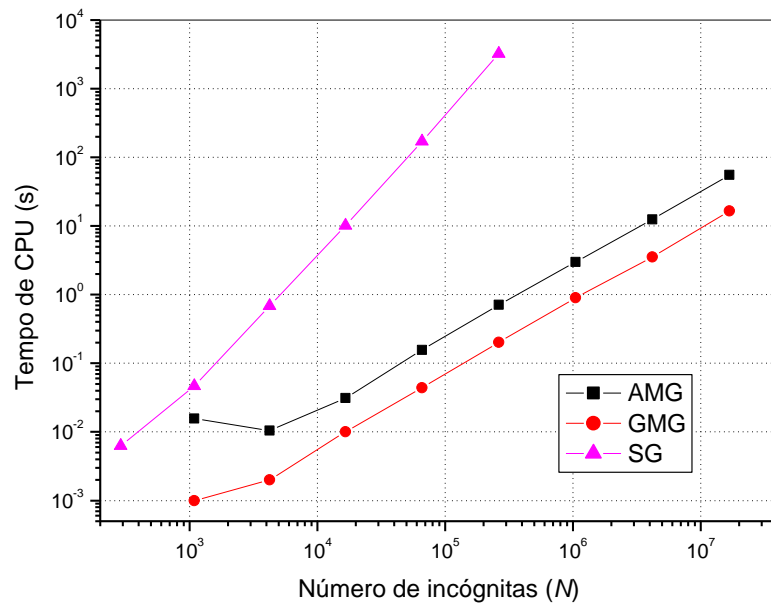


Figura 4.5: Efeito do número de incógnitas no tempo de CPU para o problema linear em malhas quadrangulares

O comportamento apresentado pelo AMG, GMG e SG com o aumento do número de incógnitas é o esperado (WATANABE et al., 2005). Na Fig. 4.5 nota-se que as curvas que representam o AMG e o GMG estão um pouco afastadas, mas possuem praticamente a mesma

inclinação. Já a curva que representa o SG, tem uma inclinação maior, além de estar mais afastada das demais.

Na Tab. 4.3 são dados os valores para o expoente p e para o coeficiente c da Eq. (3.4) para os três métodos aqui empregados. O método *multigrid* ideal é aquele em que o expoente p está próximo da unidade (WESSELING, 1992 e BRIGGS et al., 2000).

Tabela 4.3: Valores de p e c para o problema linear em malhas quadrangulares

Método empregado	Expoente p	Coeficiente c
AMG	1,05198	1,36500E-06
GMG	1,08071	2,63688E-07
SG	1,92407	2,12466E-08

Nota-se que os resultados obtidos para a equação de Laplace são os esperados. Os expoentes para o método *multigrid*, tanto algébrico como geométrico, se mantiveram próximos da unidade enquanto o expoente para o *singlegrid* ficou próximo de 2. Os coeficientes c expressam o distanciamento entre as curvas. Este resultado é complementar à Fig. 4.5.

4.1.6 Otimização do algoritmo para o AMG

Neste trabalho, o principal objetivo é a otimização do código computacional, através da análise de parâmetros previamente determinados. Estes parâmetros, testados um a um, resultam em um conjunto ótimo de dados para cada um dos casos analisados. Com este conjunto de otimizações, é obtido um ganho em tempo de CPU para a resolução do problema abordado, sendo que este ganho aumenta com o número de incógnitas. Os valores para os parâmetros empregados nos algoritmos podem ser observados na Tab. 4.4.

Tabela 4.4: Parâmetros estudados para o AMG para o problema linear em malhas quadrangulares

Parâmetro estudado	AMG padrão	AMG otimizado
Número de iterações internas (ν)	1	2
Número de níveis (L)	$L_{\text{máximo}}$	$L_{\text{máximo}}$
Fator de redução de malha (θ)	0,25	0,40
Fator de forte dependência na malha grossa (ε)	0,35	0,35

Da Tab. 4.4 nota-se que apenas dois dos quatro parâmetros em estudo tiveram seus valores modificados. Com estas modificações no algoritmo chega-se a um ganho de 11,50%

na malha mais refinada. Esta comparação entre os tempos de CPU obtidos para cada algoritmo pode ser observada na Fig. 4.6, onde nota-se que quanto maior é o número de incógnitas, maior é o ganho no tempo de CPU.

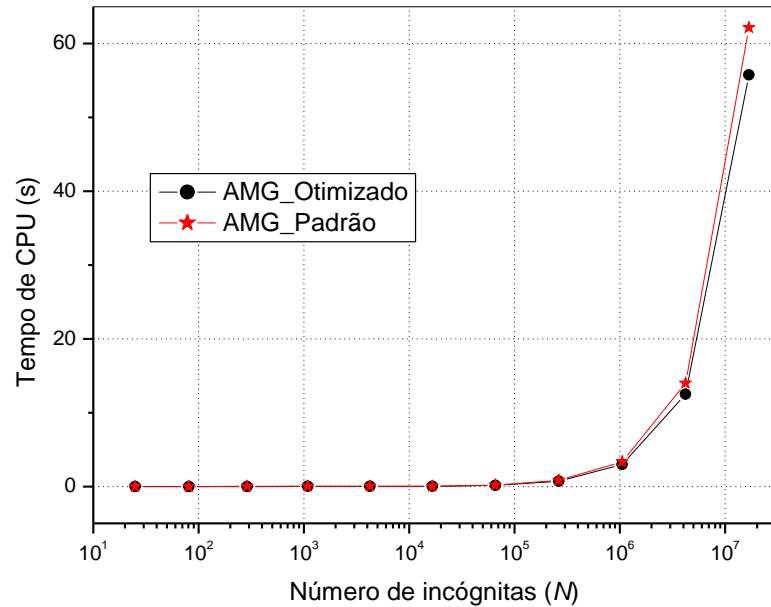


Figura 4.6: AMG otimizado *versus* AMG padrão para o problema linear em malhas quadrangulares

Com base nestes resultados, foram obtidos os expoentes p e os coeficientes c da Eq. (3.4) para o AMG padrão e o otimizado, que podem ser observados na Tab. 4.5. Além dos expoentes também são comparados o número de ciclos (iterações externas) que o AMG leva para atingir a tolerância estipulada em 10^{-8} .

Tabela 4.5 Valores de p e c para o problema linear em malhas quadrangulares

Método empregado	Expoente p	Coefficiente c	Ciclos
AMG Padrão	1,06361	1,46565E-06	10
AMG Otimizado	1,05198	1,36500E-06	6

Da Tab. 4.5 nota-se que o expoente p da Eq. (3.4) para o algoritmo otimizado está levemente mais próximo da unidade que o valor obtido com o algoritmo padrão. Observa-se também que o coeficiente c apresenta um valor ligeiramente menor para o algoritmo otimizado. O número de iterações externas (ciclos) empregadas para se atingir a tolerância desejada é menor para o algoritmo otimizado. Este valor é reduzido praticamente pela metade do valor utilizado para o algoritmo padrão.

4.1.7 Comparação entre número de ciclos para AMG e GMG

Os trabalhos que trazem comparações entre os métodos *multigrid* algébrico e geométrico, como Langer e Pusch (2006) e Wu e Elman (2006), apresentam um estudo sobre o número de ciclos utilizados pelos dois métodos, de forma a se atingir uma tolerância previamente determinada. Considerando a tolerância empregada neste trabalho, 10^{-8} , e os problemas pequeno, médio e grande, pode-se comparar a quantidade de ciclos para o AMG e o GMG através da Tab. 4.6.

Tabela 4.6: Número de ciclos para AMG e GMG para o problema linear em malhas quadrangulares

Tamanho do problema	AMG	GMG
257x257	6	11
1025x1025	6	11
4097x4097	6	11

Como pode ser observado na Tab. 4.6, o AMG atinge a tolerância estipulada em menos ciclos V que o GMG. Estes resultados concordam com os obtidos por Langer e Pusch (2006) e Wu e Elman (2006). Observa-se também que, tanto para o AMG quanto para o GMG, não importa o tamanho do problema que está sendo resolvido, o número de ciclos é sempre o mesmo.

4.2 Equação de Laplace – Problema senoidal

A equação de Laplace é resolvida novamente em um domínio de cálculo quadrado, com condição de contorno dada por uma função senoidal ao norte. O objetivo desta etapa é obter os parâmetros ótimos para esta equação, sendo que os resultados aqui obtidos são comparados aos obtidos para o problema linear. São comparados também aos resultados de Oliveira et al. (2008), que resolvem esta mesma equação empregando o método *multigrid* geométrico.

Como na seção anterior, os parâmetros a serem analisados são: número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ), fator de forte dependência na malha grossa (ε) e efeito do número de incógnitas no tempo de CPU. Depois de definir as componentes algorítmicas ótimas, chega-se a um algoritmo otimizado. O algoritmo com as componentes indicadas pela literatura é o algoritmo padrão. Para estes dois algoritmos são

comparados os tempos de CPU, o expoente p e o coeficiente c da Eq. (3.4), e ainda a quantidade de iterações externas empregadas até que se atinja a tolerância desejada.

4.2.1 Número de iterações internas (ν)

Para se avaliar a influência do número de iterações internas no tempo no CPU são feitos testes com este parâmetro variando entre 1 e 20 para os problemas considerados pequeno, médio e grande. Para estes testes computacionais são considerados todos os níveis de malha, fator de redução de malha $\theta = 0,25$ e fator de forte dependência na malha grossa $\varepsilon = 0,35$, ou seja, os parâmetros comumente empregados na literatura. Os resultados obtidos podem ser observados na Fig. 4.7.

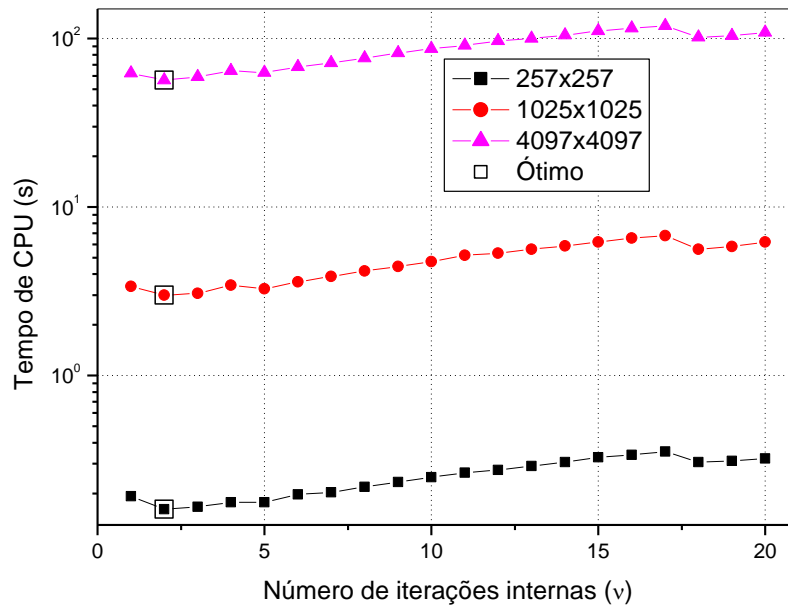


Figura 4.7: Tempo de CPU *versus* número de iterações internas para o problema senoidal em malhas quadrangulares

Conforme pode ser observado na Fig. 4.7 existe um número de iterações internas que resulta no menor tempo de CPU para os três tamanhos de problema analisados. Para o problema senoidal, o número ótimo de iterações internas é 2, ou seja, $\nu_{\text{ótimo}} = 2$. Observa-se também que, quanto maior o número de iterações internas, maior é o tempo de CPU gasto para se resolver o problema.

Em Ruge e Stüben (1986), Krechel e Stüben (1999), Briggs et al. (2000), Trottenberg et al. (2001), Iwamura et al. (2003) e Falgout (2006) é empregado $\nu = 1$ para a resolução dos

problemas. Para o problema senoidal, observou-se que o valor de $\nu_{\text{ótimo}}$ é o mesmo que o obtido para o método *multigrid* geométrico em Oliveira et al. (2008). Este valor também coincide com o encontrado para o problema linear resolvido com AMG.

O ganho em tempo de CPU fica em 19,4%, 12,8% e 9,9%, para os problemas considerados pequeno, médio e grande respectivamente, se comparado ao valor amplamente empregado pela literatura, $\nu = 1$. Com base nas conclusões acima, o valor a ser empregado para o número de iterações internas nos próximos testes computacionais, para o problema senoidal, é $\nu_{\text{ótimo}} = 2$.

4.2.2 Número de níveis (L)

Para o estudo do efeito do número de níveis no tempo de CPU é considerado o valor ótimo encontrado para número de iterações internas, $\nu = 2$. Os demais parâmetros são mantidos: para o fator de redução de malha, $\theta = 0,25$ e para o fator de forte dependência na malha grossa, $\varepsilon = 0,35$.

Os valores empregados para o número de níveis são variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia com o tamanho do problema, ou seja, $L_{\text{máximo}}, L_{\text{máximo}} - 1, L_{\text{máximo}} - 2$, e assim sucessivamente. Para a determinação deste valor para o número de níveis, deve-se observar o crescimento do tempo de CPU, com a diminuição do número de níveis. O valor para L tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele em que a resolução do problema leva em torno de 3 horas. Deve-se observar que com este valor adotado já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar. Se fossem feitos testes com outros valores para L (menores que o mínimo aqui adotado), os tempos de CPU resultantes estariam entre 20 e 24 horas, valores que dificultariam a visualização gráfica deste parâmetro. Uma análise deste parâmetro, para os três tamanhos de problema analisados neste trabalho, podem ser observados na Fig. 4.8.

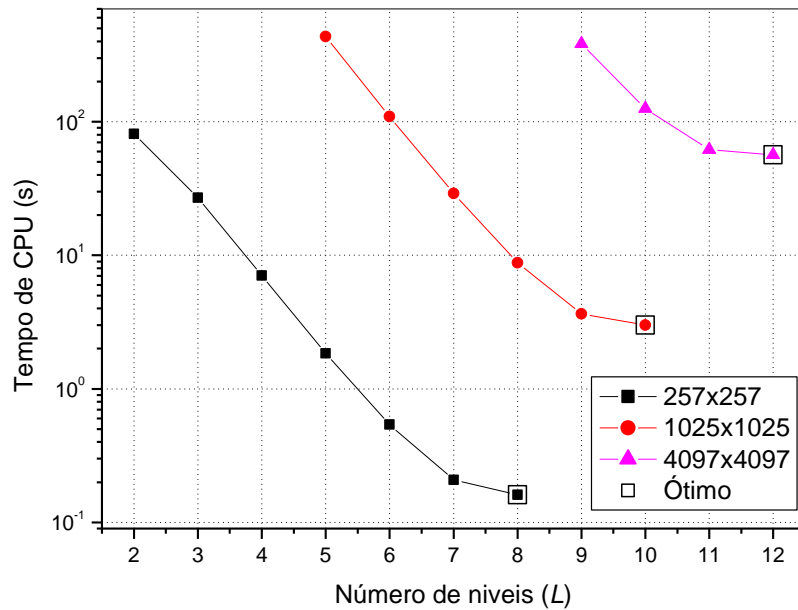


Figura 4.8: Tempo de CPU *versus* número de níveis para o problema senoidal em malhas quadrangulares

Conforme pode ser observado na Fig. 4.8, existe um número de níveis que resulta no menor tempo de CPU. Este número é o valor máximo de níveis que pode ser empregado para cada tamanho de problema, ou seja, para $L_{\text{máximo}} = 8$, tem-se $L_{\text{ótimo}} = 8$, para $L_{\text{máximo}} = 10$, tem-se $L_{\text{ótimo}} = 10$ e para $L_{\text{máximo}} = 12$ tem-se $L_{\text{ótimo}} = 12$. Desta análise pode-se concluir que existe um número ótimo de níveis, que resulta no menor tempo de CPU. Este valor é o número máximo de níveis, que varia conforme o tamanho do problema. Desta forma, o valor a ser empregado nas próximas análises é $L_{\text{ótimo}} = L_{\text{máximo}}$. Pode-se concluir também que o tamanho do problema não exerce influência sobre o valor ótimo para o número de níveis. Da Fig. 4.8, observa-se também que, quanto menor o número de níveis de malha empregados, maior é o tempo de CPU gasto para se resolver o problema.

Os resultados apresentados acima concordam com os obtidos por Oliveira et al. (2008), onde é recomendado o uso de todos os níveis de malha na resolução do problema senoidal, com o uso do GMG. Estes resultados também concordam com os descritos anteriormente para o problema linear resolvido com o AMG. Durante a revisão bibliográfica, observou-se que alguns autores, como Ruge e Stüben (1986), Falgout (2006), Langer e Pusch (2006) não se preocuparam com o número de malhas a ser empregado. Nos trabalhos em que este dado está explícito, nota-se a utilização de todos os níveis de malha, como por exemplo em Krechel e Stüben (1999), Iwamura et al. (2003) e em Wu e Elman (2006). Não foram encontrados estudos a respeito do efeito do número de malhas no tempo de CPU para o AMG.

4.2.3 Fator de redução de malha (θ)

Para o estudo do efeito do fator de redução de malha no tempo de CPU são considerados os parâmetros ótimos já encontrados nos testes acima. Para o número de iterações internas é considerado $\nu = 2$, e para o número de malhas é empregado $L = L_{\text{máximo}}$. Para o problema senoidal, os testes computacionais realizados para o problema linear são aqui repetidos. Os valores testados possuem correspondência com as razões de engrossamento do GMG, além de serem empregados na literatura. O comportamento do fator de redução de malha com a variação do número de incógnitas pode ser observado na Fig. 4.9.

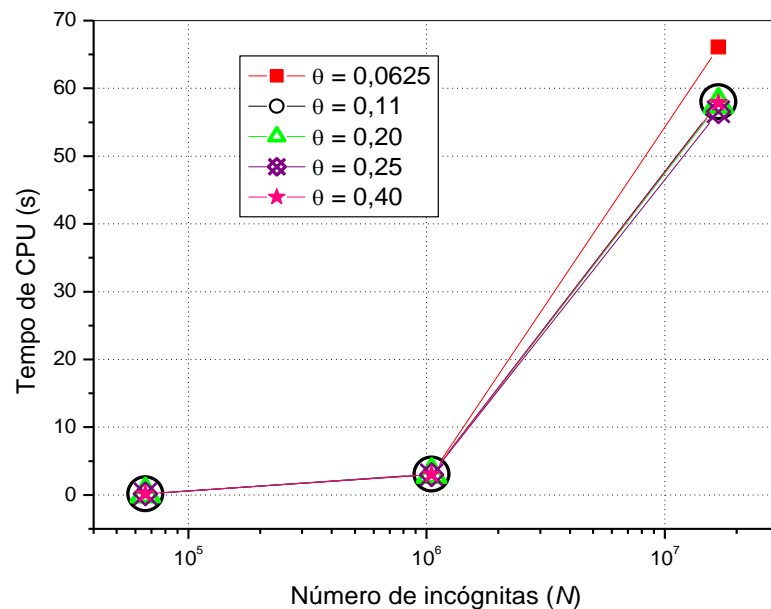


Figura 4.9: Influência do fator de redução de malha no tempo de CPU para o problema senoidal em malhas quadrangulares

Nesta figura, o fator de redução de malha que apresenta o melhor desempenho é $\theta = 0,25$, que é o valor considerado padrão pela literatura. De acordo com os resultados apresentados para o problema linear, também são obtidas as porcentagens do ganho no tempo de CPU para os diversos valores testados para θ , conforme Tab. 4.7.

O resultado apontado pela Tab. 4.7 concorda com o apresentado na Fig. 4.9, onde, na média, o valor ótimo para o fator de redução de malha é $\theta = 0,25$. Para este valor, a porcentagem que expressa a diferença entre os valores aplicados e o ótimo diminui com o refino da malha, além de ser o valor ótimo para a malha mais refinada. Desta forma, para as análises a serem feitas a seguir, emprega-se este valor, $\theta = 0,25$, que é o valor que apresenta o

melhor desempenho médio. Este valor difere do encontrado para o problema linear, que foi $\theta = 0,40$. Portanto, pode-se concluir que este é um parâmetro problema dependente. Nota-se também que o valor que apresenta o melhor desempenho médio para problema em questão coincide com o valor que é amplamente empregado na literatura.

Tabela 4.7: Diferenças entre os tempos de CPU para θ considerando o problema senoidal em malhas quadrangulares

θ	257x257	1025x1025	4097x4097
0,0625	-----	1,0%	16,9%
0,11	6,7%	1,6%	2,6%
0,2	16,7%	1,2%	1,9%
0,25	3,3%	-----	-----
0,4	3,3%	0,9%	2,3%

4.2.4 Fator de forte dependência na malha grossa (ε)

Para a análise deste parâmetro são considerados os valores ótimos encontrados para o número de iterações internas, $\nu = 2$, o número de malhas, $L = L_{\text{máximo}}$ e o fator de redução de malha, $\theta = 0,25$. Os testes feitos para este parâmetro seguem a mesma metodologia aplicada para o problema linear. Os resultados obtidos para o fator de forte dependência na malha grossa com a variação do tamanho do problema podem ser observados na Fig. 4.10.

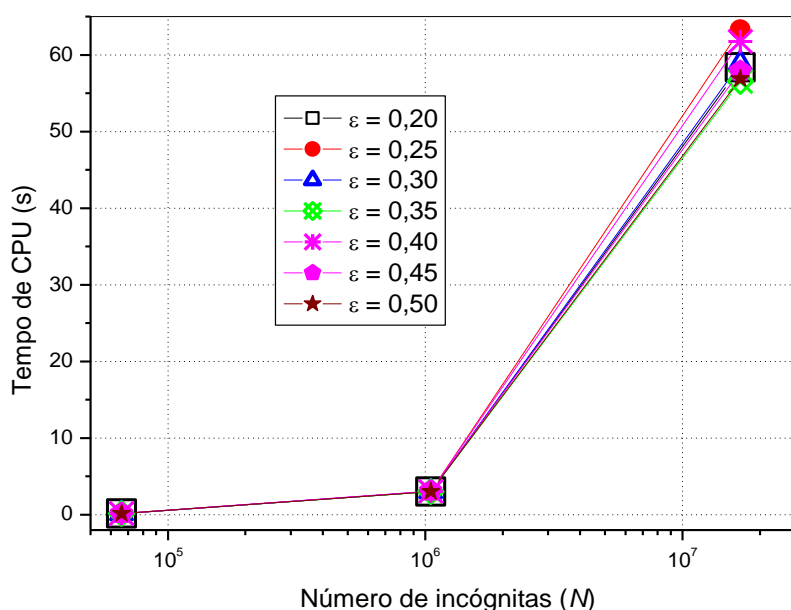


Figura 4.10: Influência do fator de forte dependência na malha grossa no tempo de CPU para o problema senoidal em malhas quadrangulares

Da observação desta figura, pode-se concluir que os tempos de CPU obtidos para alguns valores de ε , como por exemplo, $\varepsilon = 0,35$, $\varepsilon = 0,45$, $\varepsilon = 0,50$ estão muito próximos. Por este motivo, para a determinação do valor ótimo para este parâmetro, são obtidas as porcentagens do ganho no tempo de CPU dos diversos valores testados para ε , conforme pode ser observado na Tab. 4.8.

Tabela 4.8: Diferenças entre os tempos de CPU para ε considerando o problema senoidal em malhas quadrangulares

ε	257x257	1025x1025	4097x4097
0,20	3,4%	0,9%	3,2%
0,25	6,9%	0,7%	12,0%
0,30	-----	0,9%	4,1%
0,35	6,9%	-----	-----
0,40	10,3%	1,4%	9,2%
0,45	10,3%	2,6%	2,3%
0,50	10,3%	0,9%	0,6%

Com a análise da Tab. 4.8, pode-se concluir que o valor que apresenta o melhor desempenho médio é $\varepsilon = 0,35$, pois para este valor, nota-se que a porcentagem que expressa a diferença entre os valores aplicados e o ótimo diminui com o refino da malha, além de ser o valor ótimo para as duas maiores malhas testadas. Com base neste estudo, pode-se concluir que o fator de forte dependência na malha grossa a ser empregado nas próximas simulações é $\varepsilon = 0,35$. Este valor coincide com o encontrado para o problema linear, além de ser o valor que é amplamente empregado na literatura.

4.2.5 Número de incógnitas (N)

Para se avaliar o efeito do número de incógnitas sobre o tempo de CPU são considerados todos os parâmetros ótimos já encontrados para o problema senoidal. Os parâmetros são: número de iterações internas $\nu = 2$, número de malhas $L = L_{máximo}$, fator de redução de malha $\theta = 0,25$ e fator de forte dependência na malha grossa $\varepsilon = 0,35$. Para estes testes são resolvidos problemas com o número de incógnitas variando de 5x5 até 4097x4097.

Os resultados obtidos para o AMG são comparados aos obtidos para o SG e para o GMG, conforme Fig. 4.11. Nesta figura pode ser observado o comportamento do tempo de CPU com a variação do número de incógnitas. Nota-se que é um comportamento muito

semelhante ao apresentado na Fig. 4.5, quando esta mesma análise é feita para o problema linear. Pode-se notar também que as curvas que correspondem ao AMG e ao GMG possuem quase a mesma inclinação, que é diferente para o SG. Percebe-se ainda que o GMG é mais rápido que o AMG, que por sua vez é mais rápido que o SG.

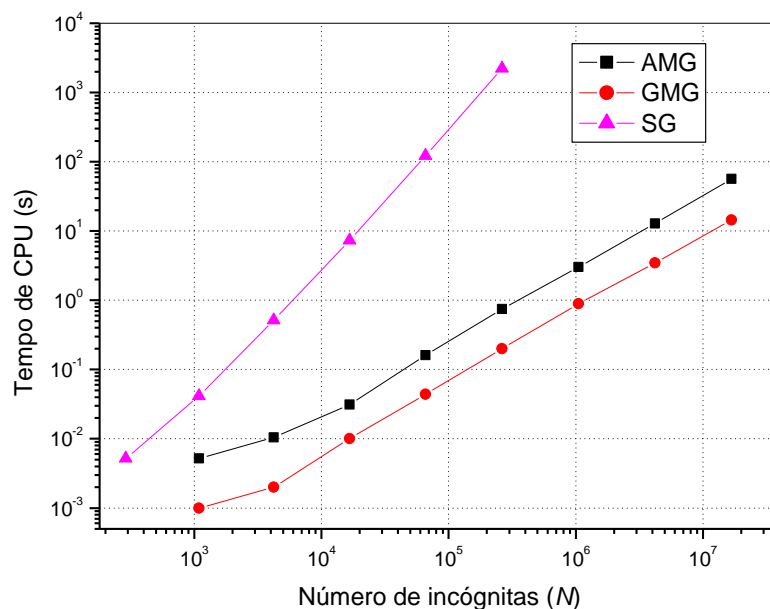


Figura 4.11: Efeito do número de incógnitas no tempo de CPU para o problema senoidal em malhas quadrangulares

Um estudo a respeito da inclinação das curvas apresentadas na Fig. 4.11 é feito com o uso do ajuste de curvas pelo método dos mínimos quadrados. Este ajuste é feito por uma curva geométrica, conforme a Eq. (3.4). Os resultados obtidos para esta análise podem ser observados na Tab. 4.9, que traz os valores para o expoente p e para o coeficiente c , dos três métodos aqui empregados para a resolução do problema senoidal.

Tabela 4.9: Valores de p e c para o problema senoidal em malhas quadrangulares

Método empregado	Expoente p	Coefficiente c
AMG	1,00085	2,83500E-06
GMG	1,06874	2,98855E-07
SG	1,91452	9,68345E-09

Pela Tab. 4.9 pode-se concluir que os expoentes p obtidos para as curvas ilustradas na Fig. 4.11 são os esperados. Os expoentes obtidos para o AMG e GMG estão próximos da unidade e próximos entre si, o que explica o fato das duas curvas terem praticamente a mesma inclinação. Para o SG o valor se aproxima de 2, fato esperado. Os coeficientes c expressam o

distanciamento entre as curvas, onde nota-se que a diferença entre o AMG e o GMG é menor. Estes resultados estão de acordo com os obtidos para o problema linear.

4.2.6 Otimização do algoritmo para o AMG

Nesta seção são apresentados os ganhos em tempo de CPU obtidos para o problema senoidal. Da mesma forma que na seção anterior, são comparados os tempos de CPU obtidos para o problema resolvido com o algoritmo padrão e o otimizado. O algoritmo otimizado é aquele em que os parâmetros ótimos encontrados a partir dos testes computacionais são os empregados na resolução do problema. Este conjunto de parâmetros ótimos, quando aplicados ao problema que está sendo resolvido, gera um ganho em tempo de CPU. Observa-se que este ganho aumenta com o número de incógnitas. Os valores empregados para os parâmetros dos algoritmos padrão e otimizado são dados na Tab. 4.10.

Tabela 4.10: Parâmetros estudados para o AMG para o problema senoidal em malhas quadrangulares

Parâmetro estudado	AMG padrão	AMG otimizado
Número de iterações internas (ν)	1	2
Número de níveis (L)	$L_{\text{máximo}}$	$L_{\text{máximo}}$
Fator de redução de malha (θ)	0,25	0,25
Fator de forte dependência na malha grossa (ε)	0,35	0,35

Da observação da Tab. 4.10 pode-se notar que apenas um dos parâmetros em estudo apresenta um valor ótimo diferente do aplicado no algoritmo padrão. Apesar desta única modificação no algoritmo, o ganho no tempo de CPU é de 11,50% na malha com mais incógnitas. A comparação para os tempos de CPU entre o algoritmo padrão e o otimizado pode ser observada na Fig. 4.12, onde nota-se que quanto maior o número de incógnitas, maior é a diferença entre os tempos de CPU.

Para complementar este estudo são calculados os expoentes p e os coeficientes c da Eq. (3.4) para o AMG padrão e o otimizado. Outra comparação a ser feita entre os dois algoritmos é para o número de ciclos (iteraões externas) que são realizados até atingir a tolerância desejada, 10^{-8} . Estas comparações são dadas na Tab. 4.11.

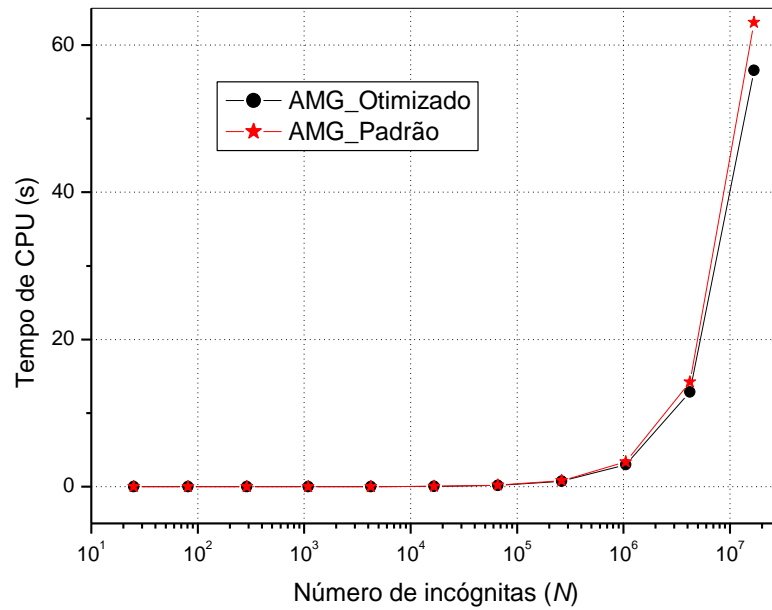


Figura 4.12: AMG otimizado *versus* AMG padrão para o problema senoidal em malhas quadrangulares

Tabela 4.11: Valores de p e c para o problema senoidal em malhas quadrangulares

Método empregado	Expoente p	Coeficiente c	Ciclos
AMG Padrão	1,05382	2,02806E-06	10
AMG Otimizado	1,00085	2,83500E-06	6

Nesta tabela, pode-se observar que o desempenho do algoritmo otimizado (que teve alteração apenas para o número de iterações internas) é bem melhor, visto que o expoente p obtido para este algoritmo está mais próximo da unidade que o expoente obtido para o algoritmo padrão. Os coeficientes c , que expressam o distanciamento entre as curvas, estão bem próximos para os dois algoritmos. Nesta mesma tabela é observado também o número de ciclos empregados para cada um dos algoritmos até atingir a tolerância determinada. Observa-se que, como foi obtido para o problema linear, o número de iterações externas obtidas para o algoritmo otimizado é menor que o obtido para o algoritmo padrão. Este número é reduzido a pouco mais da metade.

4.2.7 Comparação entre número de ciclos para AMG e GMG

Para o problema em questão também são comparados o número de ciclos (iteraões externas) para os métodos *multigrid* algébrico e geométrico. Na Tab. 4.12 tem-se um comparativo para o número de iteraões externas necessárias para se atingir a tolerância de 10^{-8} , para os problemas pequeno, médio e grande, para o AMG e GMG.

Tabela 4.12: Número de ciclos para AMG e GMG para o problema senoidal em malhas quadrangulares

Tamanho do problema	AMG	GMG
257x257	6	11
1025x1025	6	11
4097x4097	6	11

Os resultados apresentados na Tab. 4.12 concordam com os obtidos em Langer e Pusch (2006) e Wu e Elman (2006), que apresentam um estudo sobre o número de ciclos utilizados pelos dois métodos. Nestes trabalhos conclui-se que o AMG atinge a tolerância estipulada em menos ciclos que o GMG. Estes resultados também concordam com os obtidos para o problema linear. Observa-se ainda que, tanto para o AMG como para o GMG, o número de ciclos independe do tamanho do problema resolvido.

4.3 Equação de Poisson

Nesta seção são apresentados os resultados obtidos para a equação de Poisson, quando discretizada com malhas quadrangulares, em um domínio de cálculo quadrado. As condições de contorno para este problema também são do tipo Dirichlet. Com o objetivo de minimizar o tempo de CPU são feitos testes computacionais em busca dos parâmetros ótimos seguindo a mesma metodologia empregada para a equação de Laplace. Os parâmetros a serem analisados são: número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ), fator de forte dependência na malha grossa (ε) e efeito do número de incógnitas no tempo de CPU. Depois de definir as componentes algorítmicas ótimas, chega-se a um algoritmo otimizado, sendo que o algoritmo com as componentes indicadas pela literatura é o algoritmo padrão. Para estes dois algoritmos são comparados o tempo de CPU, o expoente p e coeficiente c da Eq. (3.4) e o número de iterações externas empregadas até que se atinja a tolerância desejada.

4.3.1 Número de iterações internas (ν)

Para a equação de Poisson são adotados os mesmos procedimentos já descritos anteriormente para a equação de Laplace. O número de iterações internas é variado entre 1 e

20, para os problemas pequeno, médio e grande. Para a realização destes testes computacionais são considerados todos os níveis de malha, ou seja, $L = L_{máximo}$. O valor empregado para o fator de redução de malha é $\theta = 0,25$, e para o fator de forte dependência na malha grossa é $\varepsilon = 0,35$, parâmetros amplamente usados na literatura. Na Fig. 4.13 podem ser observados os resultados obtidos para o número de iterações internas.

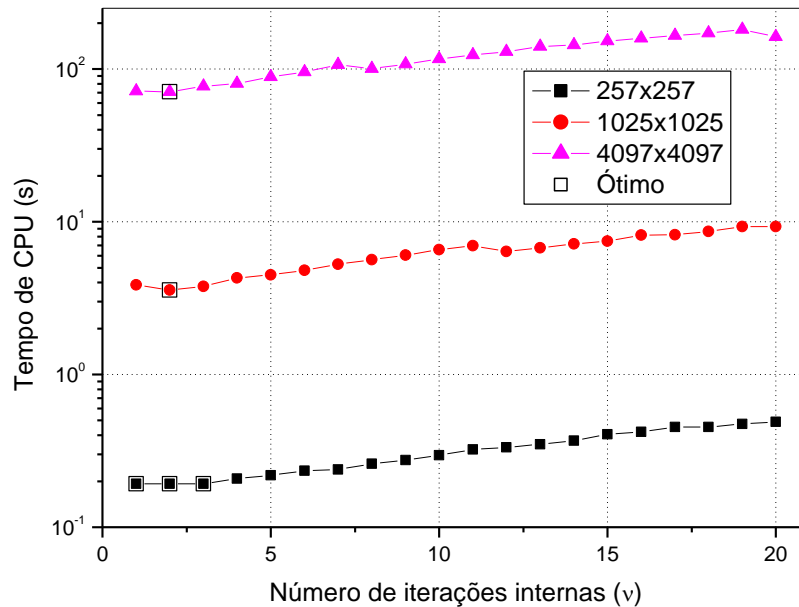


Figura 4.13: Tempo de CPU versus número de iterações internas para a equação de Poisson em malhas quadrangulares

Na Fig. 4.13 pode ser observada a existência do número ótimo de iterações internas, ou seja, o número de iterações que resulta no menor tempo de CPU. Na malha mais grosseira observa-se a existência de 3 valores que resultam no menor tempo de CPU, sendo que um deles coincide com o valor ótimo encontrado para os outros tamanhos de problema. Desta forma, pode-se concluir que o valor ótimo para o número de iterações internas quando a equação de Poisson é resolvida é $\nu_{\text{ótimo}} = 2$. Da análise da Fig. 4.13 pode-se concluir também que, quanto maior o número de iterações internas utilizadas, maior é o tempo de CPU necessário para se resolver o problema.

Nos trabalhos de Ruge e Stüben (1986), Krechel e Stüben (1999), Briggs et al. (2000), Trottenberg et al. (2001), Iwamura et al. (2003) e Falgout (2006) é empregado $\nu = 1$ para a resolução dos problemas. Para a equação de Poisson, observa-se que o valor de $\nu_{\text{ótimo}}$ é o mesmo encontrado por Oliveira et al. (2008), que resolve a equação de Laplace com o método

multigrid geométrico. Este valor também coincide com os valores ótimos encontrados com AMG para a equação de Laplace, já descritos anteriormente.

Para os problemas considerados médio e grande, obteve-se um ganho no tempo de CPU de 7,9% e 1,1%, quando os tempos obtidos para $\nu = 2$ são comparados aos obtidos para $\nu = 1$, valor que é amplamente empregado pela literatura. Baseado na discussão acima, para os próximos testes computacionais emprega-se o número ótimo de iterações internas, ou seja, $\nu_{\text{ótimo}} = 2$.

4.3.2 Número de níveis (L)

Para o estudo do efeito do número de níveis (L) no tempo de CPU é considerado o valor ótimo para o número de iterações internas, $\nu = 2$, enquanto que, para os demais parâmetros, são mantidos os mesmos valores aplicados acima: para o fator de redução de malha, $\theta = 0,25$ e para o fator de forte dependência na malha grossa, $\varepsilon = 0,35$, que são os valores indicados pela literatura.

Os valores empregados para o número de níveis são variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia com o tamanho do problema, ou seja, $L_{\text{máximo}}, L_{\text{máximo}} - 1, L_{\text{máximo}} - 2$, e assim sucessivamente. Para a determinação deste valor para o número de níveis, é observado o crescimento do tempo de CPU, com a diminuição do número de níveis. O valor para L tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele em que a resolução do problema leva em torno de 3 horas. Deve-se observar que com este valor adotado já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar. Se fossem feitos testes com outros valores para L (menores que o mínimo aqui adotado), os tempos de CPU resultantes estariam entre 20 e 24 horas, valores que dificultariam a visualização gráfica deste parâmetro.

Na Fig. 4.14 pode ser observado o comportamento deste parâmetro para os três tamanhos de problema aqui considerados. Da observação desta figura pode-se concluir que existe um número de níveis que resulta no menor tempo de CPU, para os três tamanhos de problema aqui analisados. Este valor é o número máximo de níveis que pode ser empregado em cada tamanho de problema, ou seja, para $L_{\text{máximo}} = 8$, tem-se $L_{\text{ótimo}} = 8$, para $L_{\text{máximo}} = 10$, tem-se $L_{\text{ótimo}} = 10$ e para $L_{\text{máximo}} = 12$ tem-se $L_{\text{ótimo}} = 12$. Assim, o número de níveis ótimo é o

número máximo de níveis, ou seja, $L_{\text{ótimo}} = L_{\text{máximo}}$. Pode-se concluir também que o tamanho do problema não exerce influência sobre o valor ótimo para o número de níveis.

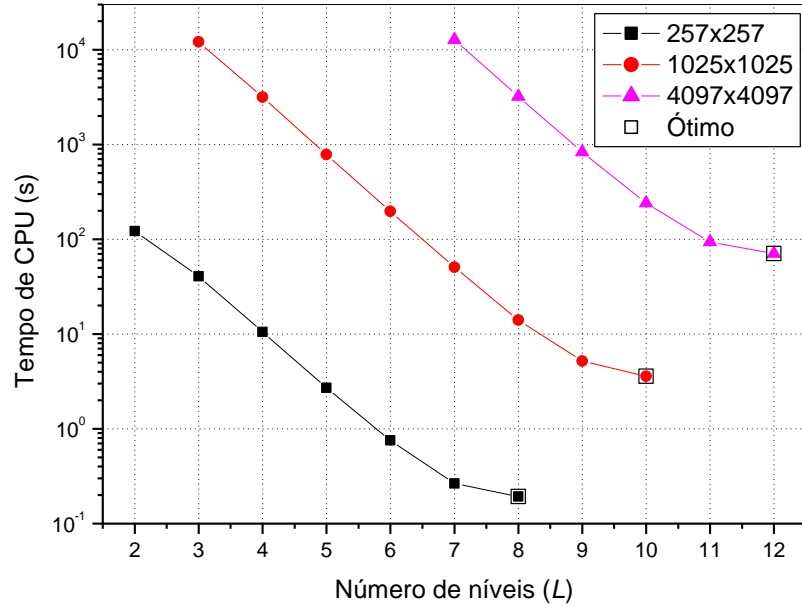


Figura 4.14: Tempo de CPU versus número de níveis para a equação de Poisson em malhas quadrangulares

Da Fig. 4.14 pode-se concluir também que, quanto menor o número de níveis de malha empregados na resolução do problema, maior é o tempo de CPU necessário para a resolução. Os resultados descritos acima concordam com os obtidos por Oliveira et al. (2008), onde é recomendado o uso de todos os níveis de malha na resolução da equação de Laplace com o uso do GMG. Estes resultados também concordam com os obtidos para a equação de Laplace com AMG, descritos anteriormente.

4.3.3 Fator de redução de malha (θ)

Para a resolução da equação de Poisson é feito o estudo do efeito do fator de redução de malha no tempo de CPU. Para estes testes são considerados os parâmetros ótimos já encontrados nos testes acima. Para o número de iterações internas é considerado $\nu = 2$, e para o número de malhas é empregado $L = L_{\text{máximo}}$. Para o fator de forte dependência na malha grossa é considerado $\varepsilon = 0,35$. A metodologia empregada para se encontrar o valor ótimo para o fator de redução de malha é a mesma empregada para a equação de Laplace. Na Fig.

4.15 pode ser observado o comportamento do fator de redução de malha com a variação do número de incógnitas.

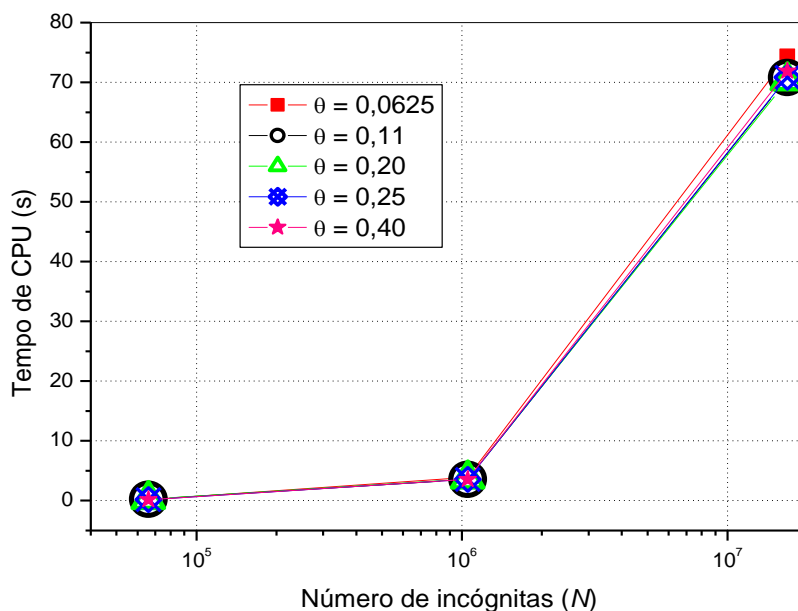


Figura 4.15: Influência do fator de redução de malha no tempo de CPU para a equação de Poisson em malhas quadrangulares

Pela observação da Fig. 4.15, não fica claro qual valor apresenta o melhor desempenho, pois os tempos de CPU obtidos para todos os valores de θ estão muito próximos. Para se determinar o valor a ser empregado nas próximas simulações, o mesmo procedimento adotado para a equação de Laplace é aplicado à equação de Poisson: são obtidas as porcentagens do ganho no tempo de CPU dos diversos valores testados para θ , conforme Tab. 4.13.

Tabela 4.13: Diferenças entre os tempos de CPU para θ considerando a equação de Poisson em malhas quadrangulares

θ	257x257	1025x1025	4097x4097
0,0625	2,9%	10,1%	5,8%
0,11	20,0%	-----	0,7%
0,20	11,4%	2,8%	-----
0,25	5,7%	0,6%	0,8%
0,40	-----	0,6%	2,3%

Da observação da Tab. 4.13 pode-se concluir que o valor a ser empregado nas próximas simulações para o fator de redução de malha é $\theta = 0,20$. Observa-se que, para este valor, a porcentagem que expressa a diferença entre os valores aplicados e o ótimo diminui com o refino da malha, além de ser o valor ótimo para a malha mais refinada. Já para o valor

amplamente empregado na literatura, $\theta = 0,25$, observa-se que primeiramente há uma diminuição nas porcentagens, para depois ser observado um crescimento com o número de incógnitas. Como o objetivo dos trabalhos em engenharia é a resolução de problemas em malhas cada vez mais refinadas (com mais incógnitas), o valor a ser empregado nas próximas simulações é $\theta = 0,20$, pois é o que apresenta o melhor desempenho médio. Apenas em Briggs et al. (2000) é empregado $\theta = 0,20$, quando é resolvida a equação de Poisson com o uso do AMG. O valor ótimo encontrado para a equação de Poisson difere dos valores encontrados para a equação de Laplace em malhas quadrangulares (problema linear: $\theta = 0,40$ e problema senoidal: $\theta = 0,25$).

4.3.4 Fator de forte dependência na malha grossa (ε)

A análise deste parâmetro para a equação de Poisson segue a mesma metodologia já empregada para a equação de Laplace. Para a realização dos testes são utilizados os valores ótimos obtidos nos estudos acima descritos. O número de iterações internas empregado é $\nu = 2$, o número de níveis de malha é o máximo possível, ou seja, $L = L_{\text{máximo}}$, e o fator de redução de malha é $\theta = 0,20$. Na Fig. 4.16 pode ser observada a influência do fator de forte dependência na malha grossa no tempo de CPU com a variação do número de incógnitas.

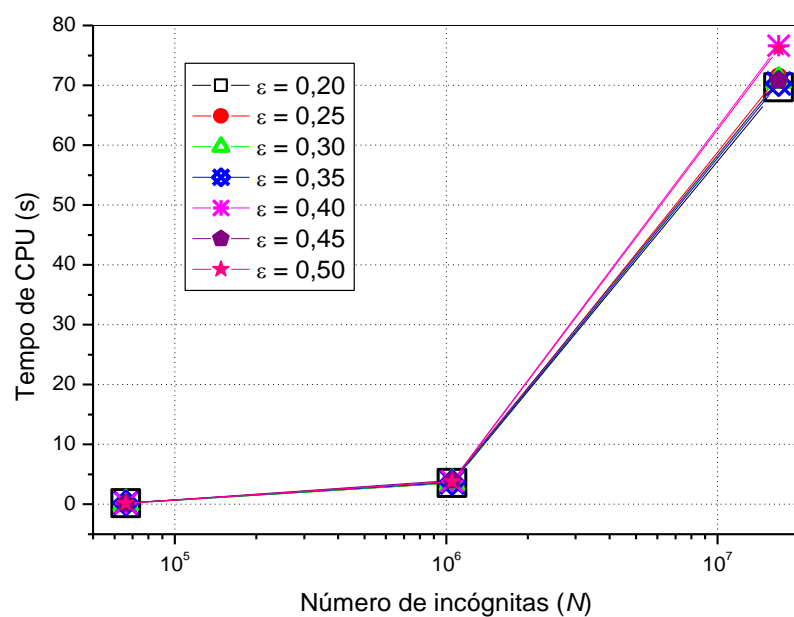


Figura 4.16: Influência do fator de forte dependência na malha grossa no tempo de CPU para a equação de Poisson em malhas quadrangulares

Como pode ser observado na Fig. 4.16, os tempos de CPU apresentados para cada valor do fator de forte dependência na malha grossa estão muito próximos entre si, exceto para dois valores que apresentam tempos um pouco maiores, $\varepsilon = 0,30$ e $\varepsilon = 0,50$. Desta forma, para a obtenção do valor com o melhor desempenho médio para este parâmetro é empregado o cálculo das porcentagens que expressam a diferença entre o valor em questão e o valor ótimo obtido. Esta análise pode ser observada na Tab. 4.14.

Tabela 4.14: Diferenças entre os tempos de CPU para ε considerando a equação de Poisson em malhas quadrangulares

ε	257x257	1025x1025	4097x4097
0,20	-----	-----	-----
0,25	11,1%	2,1%	2,7%
0,30	2,8%	0,3%	1,6%
0,35	17,1%	3,1%	0,9%
0,40	-----	8,9%	9,9%
0,45	-----	13,5%	1,6%
0,50	-----	7,2%	9,2%

Como pode ser observado na Tab. 4.14, o valor a ser considerado ótimo é $\varepsilon = 0,20$. Este é o valor que obteve o menor tempo de CPU nos três tamanhos de problema analisados. Desta forma, pode-se concluir que o valor ótimo para este parâmetro é $\varepsilon_{\text{ótimo}} = 0,20$. Nota-se também que o valor considerado padrão na literatura, $\varepsilon = 0,35$, não apresenta o melhor desempenho dentre os valores testados. O resultado aqui encontrado coincide com o disponível em Trottenberg et al. (2001), que também resolve a equação de Poisson e aplica $\varepsilon = 0,20$. Com base nesta análise, pode-se concluir que o valor a ser empregado nas próximas simulações, para o fator de forte dependência na malha grossa é $\varepsilon_{\text{ótimo}} = 0,20$.

4.3.5 Número de incógnitas (N)

Nesta etapa do trabalho são apresentados os resultados a respeito do efeito do número de incógnitas no tempo de CPU para o AMG. Para esta análise são considerados os valores ótimos encontrados, ou seja, para o número de iterações internas é empregado $\nu = 2$, o número de malhas utilizado é o máximo possível ($L = L_{\text{máximo}}$), o fator de redução de malha é $\theta = 0,20$ e o fator de forte dependência na malha grossa é $\varepsilon = 0,20$. A metodologia empregada neste estudo é a mesma utilizada para a equação de Laplace, ou seja, são

resolvidos problemas onde o número de incógnitas está variando de 5×5 até 4097×4097 . Os resultados obtidos para o AMG também são comparados aos obtidos para SG e GMG, conforme pode ser observado na Fig. 4.17.

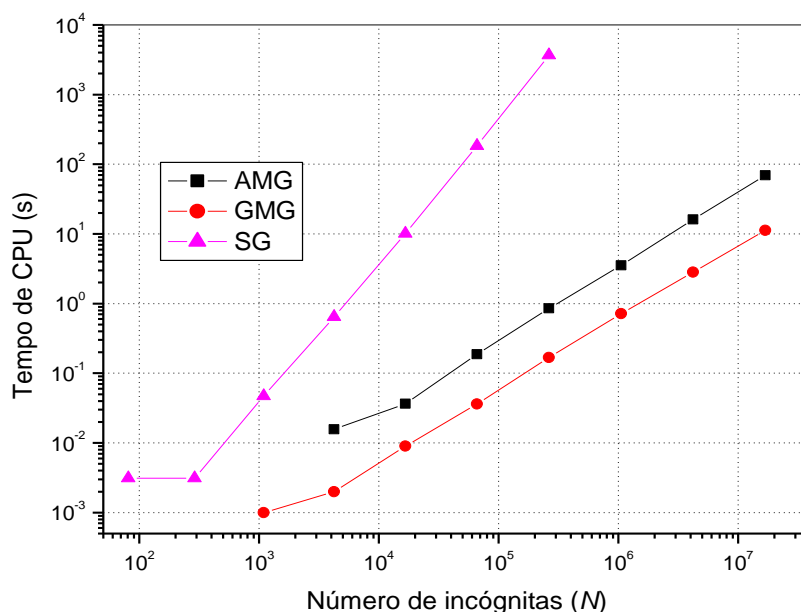


Figura 4.17: Efeito do número de incógnitas no tempo de CPU para a equação de Poisson em malhas quadrangulares

Conforme pode ser observado na Fig. 4.17, o comportamento obtido para as curvas que representam o AMG, GMG e SG é o esperado. Estes resultados estão de acordo com os encontrados para a equação de Laplace, descritos nas seções anteriores. Concordam também com os resultados obtidos por Cleary et al. (2000), que resolvem a equação de Poisson e trazem um gráfico que ilustra o crescimento do tempo de CPU com o aumento do número de incógnitas. Da análise da Fig. 4.17 pode-se concluir que as curvas correspondentes ao AMG e GMG possuem praticamente a mesma inclinação. Já para a curva que representa o SG, a inclinação é diferente. Observa-se ainda que, o GMG é mais rápido que o AMG, que, por sua vez é mais rápido que o SG.

Um estudo adicional a respeito das inclinações das curvas da Fig. 4.17 é apresentado. Este estudo é feito com o uso do ajuste de curvas pelo método dos mínimos quadrados, conforme Eq. (3.4). Os expoentes p e os coeficientes c obtidos para o AMG, GMG e SG, podem ser observados na Tab. 4.15.

Tabela 4.15: Valores de p e c para a equação de Poisson em malhas quadrangulares

Método empregado	Expoente p	Coefficiente c
AMG	1,04212	1,93200E-06
GMG	1,04387	3,45772E-07
SG	2,04112	2,78535E-08

Conforme os dados apresentados na Tab. 4.15, conclui-se que os expoentes p obtidos para as curvas da Fig. 4.17 estão de acordo com o esperado. Os coeficientes c para cada um dos métodos ilustram o distanciamento entre as curvas na Fig. 4.17. Para o AMG e o GMG os valores estão próximos da unidade e entre si. Para o SG este valor está próximo de 2. Estes resultados concordam com os obtidos para a equação de Laplace, descritos nas seções anteriores.

4.3.6 Otimização do algoritmo para o AMG

Com os parâmetros ótimos obtidos nas seções anteriores chega-se a um algoritmo ótimo para ser empregado na resolução da equação de Poisson com o emprego do AMG. Este algoritmo apresenta ganhos em termos de tempo de CPU e também em número de iterações externas (ciclos V) quando comparado ao algoritmo padrão. Os valores considerados para o algoritmo padrão e para o otimizado podem ser observados na Tab. 4.16.

Tabela 4.16: Parâmetros estudados para o AMG para a equação de Poisson em malhas quadrangulares

Parâmetro estudado	AMG padrão	AMG otimizado
Número de iterações internas (ν)	1	2
Número de níveis (L)	$L_{\text{máximo}}$	$L_{\text{máximo}}$
Fator de redução de malha (θ)	0,25	0,20
Fator de forte dependência na malha grossa (ε)	0,35	0,20

Da Tab. 4.16 pode-se notar que apenas o número de níveis manteve-se igual para os dois algoritmos. Os outros três parâmetros têm seus valores ótimos diferentes daqueles que são amplamente empregados na literatura. Com estas modificações no algoritmo, para a malha com mais incógnitas, o ganho obtido é de 8,42% no tempo de CPU do algoritmo ótimo, quando comparado ao tempo de CPU do algoritmo padrão. A Fig. 4.18 ilustra a comparação entre os dois algoritmos considerados. Desta figura, pode-se observar que quanto maior o número de incógnitas, maior é o ganho em tempo de CPU, com o uso do algoritmo ótimo.

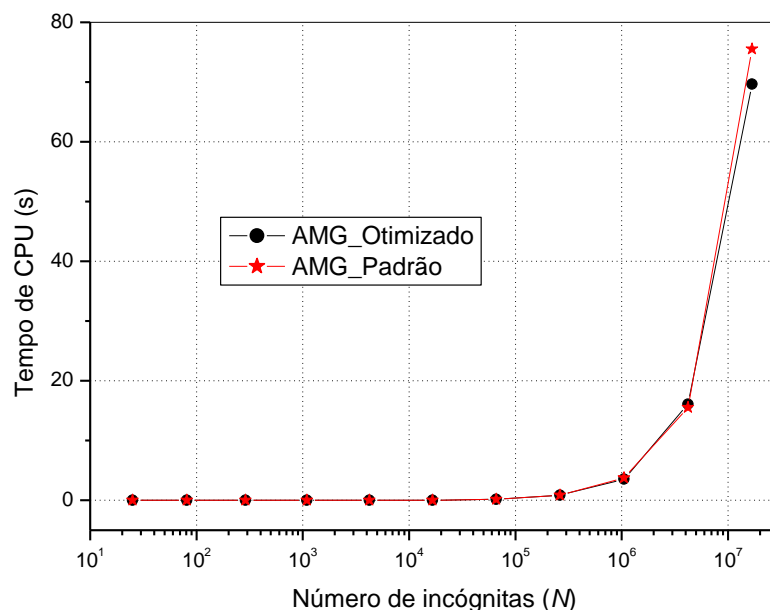


Figura 4.18: AMG otimizado *versus* AMG padrão para a equação de Poisson em malhas quadrangulares

Uma análise a respeito dos expoentes p e dos coeficientes c das curvas para o AMG padrão e o otimizado é feita através do ajuste de curvas pelo método dos mínimos quadrados. Outro item a ser comparado entre os dois algoritmos é a quantidade de iterações externas (ciclos V) que são realizadas até que se atinja a tolerância desejada, 10^{-8} . Estas comparações podem ser observadas na Tab. 4.17.

Tabela 4.17: Valores de p e c para a equação de Poisson em malhas quadrangulares

Método empregado	Expoente p	Coefficiente c	Ciclos
AMG Padrão	1,12084	1,37148E-06	13
AMG Otimizado	1,04212	1,93200E-06	9

Da Tab. 4.17 pode-se concluir que o algoritmo otimizado apresenta um desempenho melhor quando comparado ao algoritmo padrão. O expoente p do algoritmo otimizado está mais próximo da unidade, e o número de ciclos também é menor. Os coeficientes c estão próximos entre si e ilustram o distanciamento entre as curvas da Fig. 4.18. Os resultados aqui obtidos para o expoente p e para o número de ciclos concordam com os obtidos para a equação de Laplace, descritos nas seções anteriores.

4.3.7 Comparação entre número de ciclos para AMG e GMG

Para a equação de Poisson também são comparados o número de iterações externas utilizadas para se atingir o critério de parada tanto para o método *multigrid* algébrico, quanto para o método *multigrid* geométrico. Na Tab. 4.18 pode-se observar o número de ciclos empregados para os problemas pequeno, médio e grande tanto para o AMG quanto para o GMG.

Tabela 4.18: Número de ciclos para AMG e GMG para a equação de Poisson em malhas quadrangulares

Tamanho do problema	AMG	GMG
257x257	8	10
1025x1025	8	10
4097x4097	9	10

Com base nos resultados apresentados na Tab. 4.18 pode-se concluir que o AMG atinge a tolerância estipulada com um número um pouco menor de iterações externas que o GMG. Estes resultados concordam com os obtidos em Langer e Pusch (2006) e Wu e Elman (2006), que apresentam um estudo sobre o número de ciclos utilizados pelos dois métodos. Estes resultados também estão de acordo com os obtidos para a equação de Laplace em malhas quadrangulares. Observa-se ainda que, para o AMG, a variação para o número de ciclos é pequena conforme o tamanho do problema. Para o GMG, o número de ciclos é o mesmo para os três tamanhos de problema.

4.4 Comparativo para malhas quadrangulares

O objetivo desta seção é comparar os resultados obtidos para cada um dos parâmetros estudados, variando o problema resolvido. São analisados os seguintes parâmetros: número de iterações internas (ν), número de níveis de malha (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). Para a análise destes parâmetros são considerados os resultados obtidos na malha com mais incógnitas, sendo que para os outros valores de N , os resultados obtidos são análogos. Além disso, são comparados os resultados obtidos para os tempos de CPU, os expoentes p e os coeficientes c da Eq. (3.4) para o AMG e também o número de iterações externas empregadas para se atingir a tolerância desejada.

4.4.1 Número de iterações internas (ν)

Um estudo comparativo para os três problemas aqui resolvidos é feito através da Fig. 4.19.

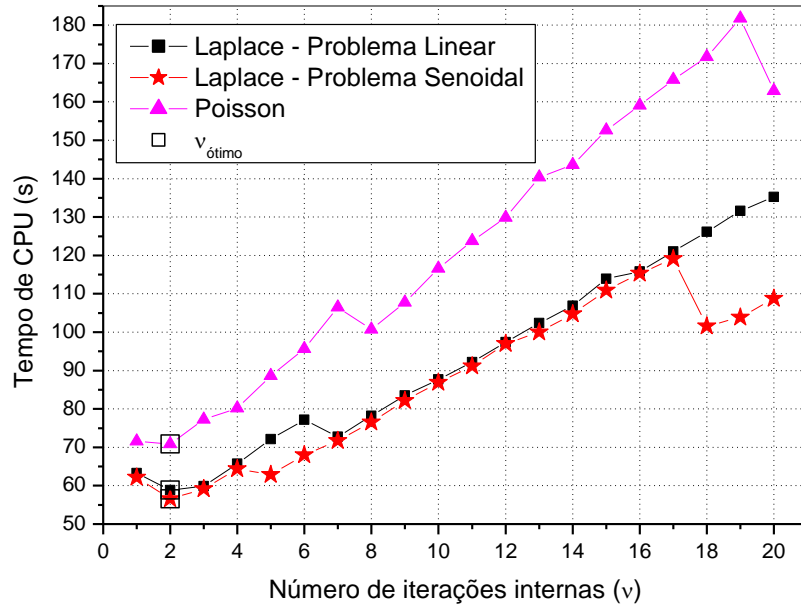


Figura 4.19: Número de iterações internas para malhas quadrangulares

Na Fig. 4.19 pode ser observado o comportamento para o número de iterações internas, com o problema resolvido na malha com mais incógnitas. Com esta análise, pode-se concluir que existe uma tendência para este parâmetro, quando o problema é resolvido em malhas quadrangulares. Para os três problemas aqui resolvidos, o valor ótimo é o mesmo, ou seja, $\nu_{\text{ótimo}} = 2$.

4.4.2 Número de níveis (L)

Um estudo comparativo para o número ótimo de níveis entre os três problemas abordados em malhas quadrangulares é aqui apresentado. Com base na Fig. 4.20, pode-se concluir que, para malhas quadrangulares o ideal é se utilizar todos os níveis possíveis de malha, ou seja, $L_{\text{ótimo}} = L_{\text{máximo}}$.

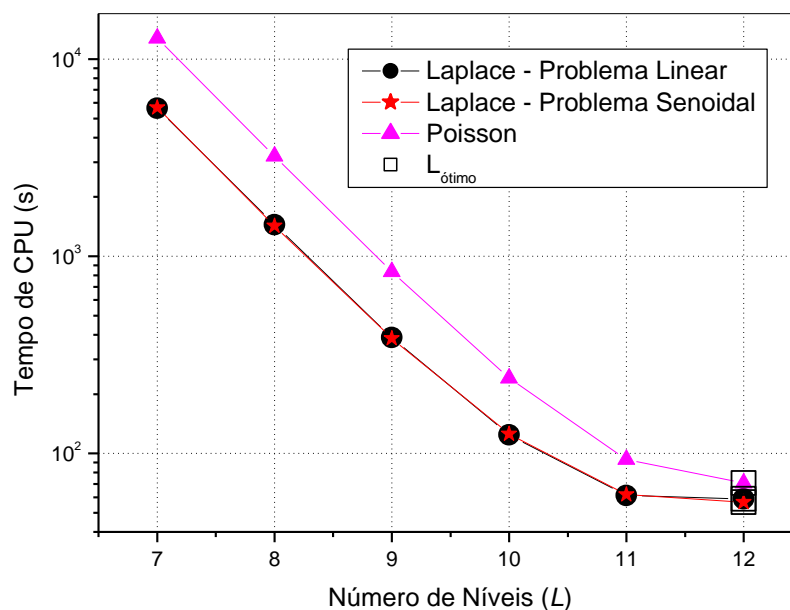


Figura 4.20: Número de níveis para malhas quadrangulares

4.4.3 Fator de redução de malha (θ)

Os valores encontrados para o fator de redução de malha em cada um dos problemas em estudo são comparados na Fig. 4.21. Da observação desta figura, pode-se concluir que o valor amplamente recomendado pela literatura ($\theta=0,25$) não é o valor que apresenta o melhor desempenho para as equações resolvidas em malhas quadrangulares. Observa-se também que são encontrados valores ótimos diferentes para os três problemas resolvidos.

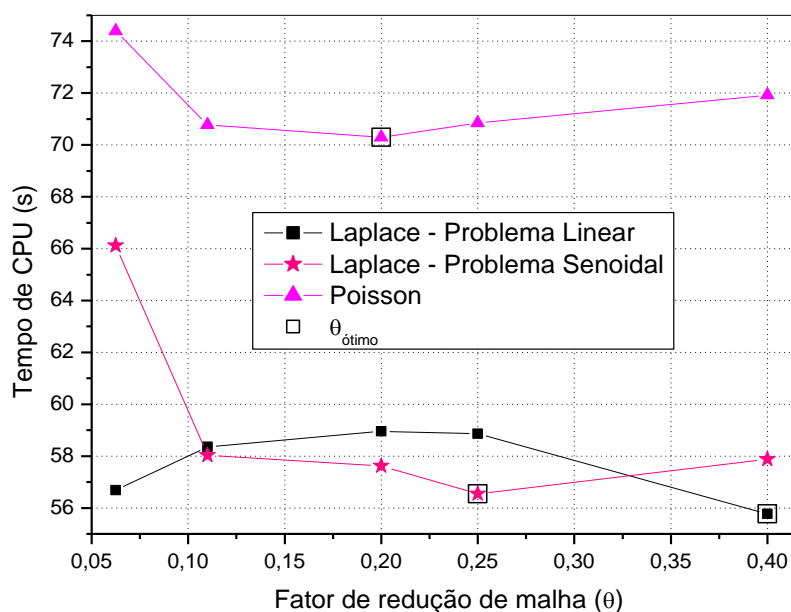


Figura 4.21: Fator de redução de malha para malhas quadrangulares

Para malhas quadrangulares, o valor empregado para θ não irá influenciar na construção das malhas auxiliares, pois todos os vizinhos estarão fortemente conectados entre si, independente do valor aplicado. A explicação para este fato pode ser observada na Fig. 4.21, onde nota-se a pequena variação para os tempos de CPU, com a variação do valor para θ , em cada uma das equações.

4.4.4 Fator de forte dependência na malha grossa (ε)

São comparados os valores encontrados para o fator de forte dependência na malha grossa em malhas quadrangulares. Estes resultados podem ser observados na Fig. 4.22.

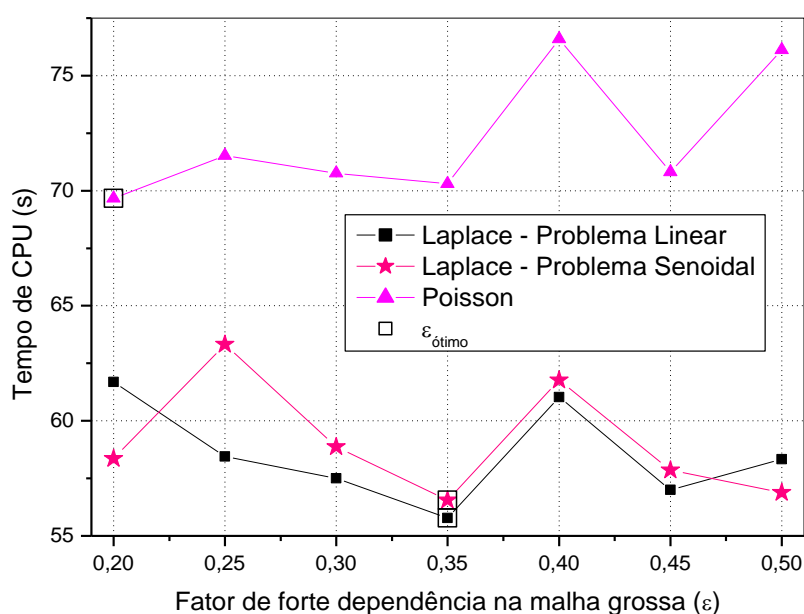


Figura 4.22: Fator de forte dependência na malha grossa para malhas quadrangulares

Nesta figura, pode-se notar que, para a equação de Laplace, o valor que apresenta o melhor desempenho é o amplamente recomendado pela literatura, $\varepsilon = 0,35$. Para a equação de Poisson este valor não apresenta o melhor desempenho, mas é muito próximo do ótimo, que é dado por $\varepsilon = 0,20$.

4.4.5 Número de incógnitas (N)

Durante a análise do efeito do número de incógnitas no tempo de CPU, é observado que um comportamento padrão é apresentado tanto para a equação de Laplace, quanto para a equação de Poisson. Em ambos os casos, o tempo de CPU aumenta quase linearmente com o aumento do número de incógnitas. É observado também que as curvas obtidas para o AMG e para o GMG mantiveram-se próximas e com inclinações parecidas. Para o SG, as curvas se mantiveram afastadas do AMG e do GMG, além de apresentarem uma inclinação maior, como observado na Fig. 4.23, que ilustra todas as curvas citadas. Um comparativo para os expoentes p e os coeficientes c da Eq. (3.4) para a equação de Laplace e Poisson pode ser observado na Tab. 4.19.

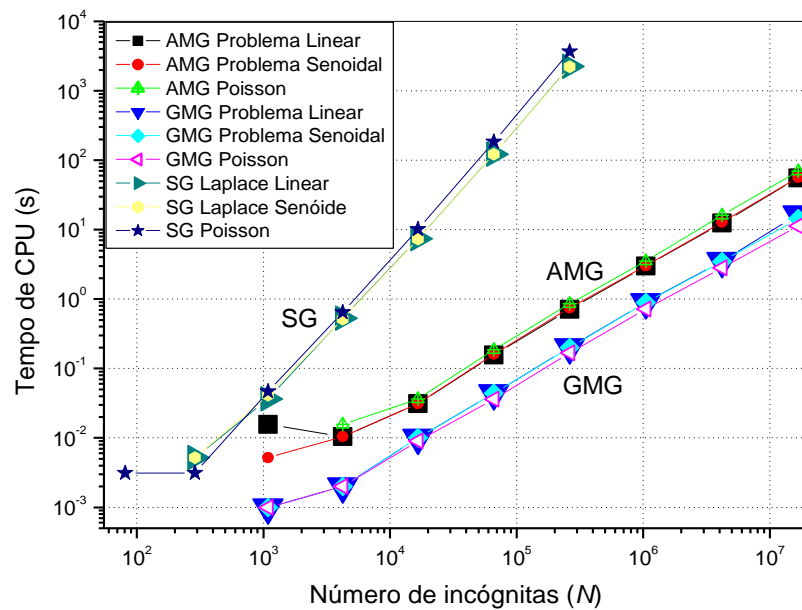


Figura 4.23: Tempo de CPU para AMG, GMG e SG para malhas quadrangulares

Tabela 4.19: Valores de p e c para malhas quadrangulares

Problema	Linear	Senoidal	Poisson
AMG (c)	1,36500E-06	2,83500E-06	1,93200E-06
GMG (c)	2,63688E-07	2,98855E-07	3,45772E-07
SG (c)	2,12466E-08	9,68345E-09	2,78535E-08
AMG (p)	1,06361	1,05382	1,04212
GMG (p)	1,08071	1,06874	1,04387
SG (p)	1,92407	1,91452	2,04112

4.4.6 Otimização do algoritmo para o AMG

Com base nos estudos realizados sobre o efeito da variação dos parâmetros do AMG no tempo de CPU, pode-se concluir que, para cada uma das equações analisadas, tem-se um conjunto de parâmetros ótimos, que irá resultar no menor tempo de CPU para a resolução dos problemas. Com a aplicação destes parâmetros ótimos é obtido um ganho em termos de tempo de CPU, representado pela coluna “Tempo”, na Tab. 4.20. A diferença entre o número de iterações externas obtidas entre o algoritmo padrão e o otimizado também pode ser observado na coluna “Ciclos” da mesma tabela. Os resultados apresentados nesta tabela são referentes à malha mais refinada.

Tabela 4.20: Ganhos obtidos com as otimizações para malhas quadrangulares

Problema	Tempo	Ciclos
Linear	11,50%	4
Senoidal	11,50%	4
Poisson	8,42%	4

4.4.7 Comparação entre número de ciclos para AMG e GMG

Os resultados obtidos para o número de ciclos referentes à resolução do AMG e do GMG, para os três problemas em questão, são qualitativamente iguais, ou seja, o AMG atinge o critério de parada com menos ciclos que o GMG, tanto para a equação de Laplace, quanto para a equação de Poisson. Estes resultados concordam com os disponíveis na literatura e podem ser observados na Tab. 4.21, onde é apresentado o número de ciclos para o AMG e para o GMG, para o problema resolvido na malha com mais incógnitas.

Tabela 4.21: Número de ciclos para AMG e GMG para malhas quadrangulares

Problema	AMG	GMG
Linear	6	11
Senoidal	6	11
Poisson	9	10

4.5 Conclusões do Capítulo 4

Neste capítulo foram obtidos os valores ótimos para o número de iterações internas, número de níveis, fator de redução de malha e fator de forte dependência na malha grossa para a equação de Poisson e para a equação de Laplace, resolvida considerando duas situações, onde são variadas as condições de contorno. Foi avaliado também o efeito do número de incógnitas no tempo de CPU, o expoente p e o coeficiente c dos métodos envolvidos na resolução das equações e o número de ciclos (iteraões externas) empregados até que o critério de parada fosse satisfeito. Comparações entre o algoritmo padrão e o otimizado também fazem parte deste capítulo. Destas análises, pode-se concluir que:

- O número de iterações internas (ν) afeta significativamente o tempo de CPU para o AMG. Para a equação de Laplace e Poisson, $\nu_{\text{ótimo}} = 2$. Na literatura, emprega-se $\nu = 1$.
- O número de níveis (L) empregados para a resolução do AMG também afeta significativamente o tempo de CPU. Para as equações acima citadas, observou-se que o número ótimo de níveis é o número máximo que pode ser empregado para cada tamanho de problema, ou seja, $L_{\text{ótimo}} = L_{\text{máximo}}$.
- O fator de redução de malha (θ), influencia pouco o tempo de CPU quando aplicado a malhas estruturadas quadrangulares. Este parâmetro variou conforme o problema que estava sendo resolvido. Para o problema linear, $\theta = 0,40$, para o problema senoidal, $\theta = 0,25$, e para Poisson, $\theta = 0,20$. Na literatura, recomenda-se usar $\theta = 0,25$ para problemas em malhas estruturadas.
- O fator de forte dependência na malha grossa (ε), influencia pouco o tempo de CPU quando aplicado a malhas estruturadas quadrangulares. Este parâmetro variou com o problema que estava sendo resolvido: tanto para o problema linear, quanto para o senoidal, $\varepsilon = 0,35$, e para Poisson, $\varepsilon = 0,20$. Na literatura, recomenda-se usar $\varepsilon = 0,35$ para problemas em malhas estruturadas.
- O efeito do número de incógnitas no tempo de CPU teve o comportamento esperado, crescendo quase linearmente com o número de incógnitas para o AMG.
- O algoritmo otimizado obteve ganhos de pelo menos 8,42% quando comparado ao algoritmo padrão, além de ser resolvido com menos iteraões externas (ciclos V).

- Os problemas resolvidos com AMG atingem a tolerância estipulada com o emprego de menos ciclos V que o GMG.
- O tempo de CPU para se resolver um problema com AMG é maior que o obtido para o GMG, para um mesmo N .

5 Resultados para malhas estruturadas triangulares

Neste capítulo são discutidos os resultados obtidos para a equação de Laplace, discretizada em malhas estruturadas triangulares, com o método dos volumes finitos, considerando um domínio de cálculo quadrado. A equação de Laplace considerada é resolvida considerando dois problemas distintos: problema linear e problema senoidal. Estes problemas já foram resolvidos em malhas quadrangulares, com o método de diferenças finitas.

São feitos testes computacionais, com a finalidade de obter um algoritmo ótimo para o AMG, quando aplicado a malhas estruturadas triangulares. Este algoritmo ótimo é composto pelos parâmetros testados, seguindo a metodologia já adotada para malhas quadrangulares. Os parâmetros em estudo são: número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). Depois de feitas as otimizações no algoritmo é avaliado o efeito do número de incógnitas (N) no tempo de CPU. São comparados os tempos de CPU para o algoritmo padrão e o otimizado, sendo avaliado o ganho obtido. É comparado também o número de iterações externas (ciclos V) necessários para se atingir a tolerância previamente determinada, para estes algoritmos e o expoente p e o coeficiente c da Eq. (3.4).

Com relação aos tamanhos de problema em malhas triangulares, é adotada a mesma metodologia empregada para malhas quadrangulares. São resolvidos problemas em três tamanhos de malha diferentes: pequeno, médio e grande, com $2^{18} = 262.144$, $2^{20} = 1.048.576$ e $2^{24} = 16.777.216$ incógnitas, respectivamente. O número de incógnitas envolve os volumes reais. Não foram encontrados na literatura trabalhos que resolvam problemas em malhas mais refinadas do que os aqui apresentados, para malhas triangulares. Para os problemas pequeno e médio, para os quais o tempo de CPU obtido é inferior a 10 segundos, as simulações são repetidas 3 vezes, de modo que os tempos de CPU (objeto de estudo deste trabalho) são dados por uma média aritmética destes valores.

Para um melhor entendimento, este capítulo está dividido em três seções: equação de Laplace - Problema linear; equação de Laplace – Problema senoidal e comparativo para malhas triangulares.

5.1 Equação de Laplace – Problema linear

Seguindo a mesma metodologia empregada para malhas quadrangulares, para esta equação são feitos testes com o objetivo de encontrar o valor ótimo para: número de iterações internas (ν), número de malhas (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). Valor ótimo é aquele que corresponde ao menor tempo de CPU para o parâmetro em questão, quando todos os outros são considerados constantes. Depois de definir as componentes algorítmicas ótimas, chega-se a um algoritmo otimizado, sendo que o algoritmo com as componentes indicadas pela literatura é o algoritmo padrão. Para estes dois algoritmos são comparados os tempos de CPU, o expoente p e o coeficiente c da Eq. (3.4), e o número de iterações externas necessárias para se atingir a tolerância desejada.

5.1.1 Número de iterações internas (ν)

Da mesma forma que no capítulo anterior, são feitos testes computacionais com o objetivo de se encontrar o valor ótimo para o número de iterações internas. Este parâmetro tem seu valor variado entre 1 e 20, conforme realizado para malhas quadrangulares. Para os demais parâmetros do AMG são considerados: número máximo de níveis, $L = L_{\text{máximo}}$, para o fator de redução de malha, $\theta = 0,25$ e para o fator de forte dependência na malha grossa, $\varepsilon = 0,35$, ou seja, os parâmetros comumente empregados na literatura.

O comportamento do número de iterações internas para os três tamanhos de problema considerados pode ser observado na Fig. 5.1. Pela análise desta figura, pode-se concluir que, independente do tamanho de problema que está sendo resolvido, existe um valor para o número de iterações internas, que resulta no menor tempo de CPU. Este valor é 1, ou seja, $\nu_{\text{ótimo}} = 1$. Observa-se ainda que, quanto maior o número de iterações internas, maior o tempo de CPU necessário para a resolução do problema. Gaspar et al. (2009), que aplica o GMG a malhas triangulares obtidas pelo método dos elementos finitos, também conclui que o valor

ótimo para ν é 1. Este valor difere do encontrado para esta mesma equação resolvida em malhas quadrangulares, ou seja, houve uma pequena diferença deste parâmetro relacionado ao tipo de malha e método. Com base na análise apresentada acima, o número de iterações internas a ser empregado nas próximas simulações é o ótimo encontrado, ou seja, $\nu_{\text{ótimo}} = 1$.

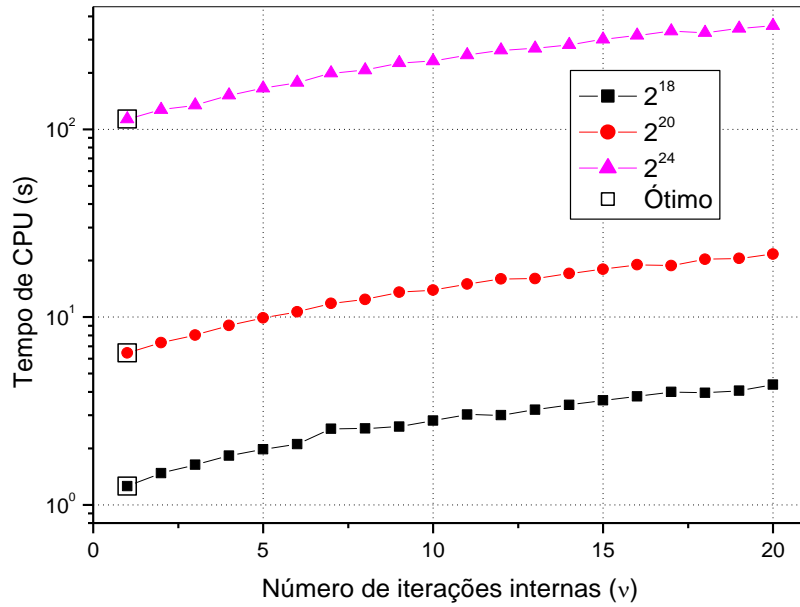


Figura 5.1: Tempo de CPU *versus* número de iterações internas para o problema linear em malhas triangulares

5.1.2 Número de níveis (L)

Para avaliar a influência do número de níveis no tempo de CPU é empregado o valor ótimo encontrado para o número de iterações internas, $\nu = 1$. Para o fator de redução de malha é utilizado $\theta = 0,25$ e para o fator de forte dependência na malha grossa, $\varepsilon = 0,35$.

Os valores empregados para o número de níveis são variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia com o tamanho do problema, ou seja, $L_{\text{máximo}}, L_{\text{máximo}} - 1, L_{\text{máximo}} - 2$, e assim sucessivamente. Para a determinação deste valor para o número de níveis, é observado o crescimento do tempo de CPU, com a diminuição do número de níveis. O valor para L tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele em que a resolução do problema leva em torno de 3 horas. Deve-se observar que com este valor adotado já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar. Se fossem feitos testes com outros valores para L

(menores que o mínimo aqui adotado), os tempos de CPU resultantes estariam entre 20 e 24 horas, valores que dificultariam a visualização gráfica deste parâmetro. Os resultados para os três tamanhos de problema podem ser observados na figura abaixo.

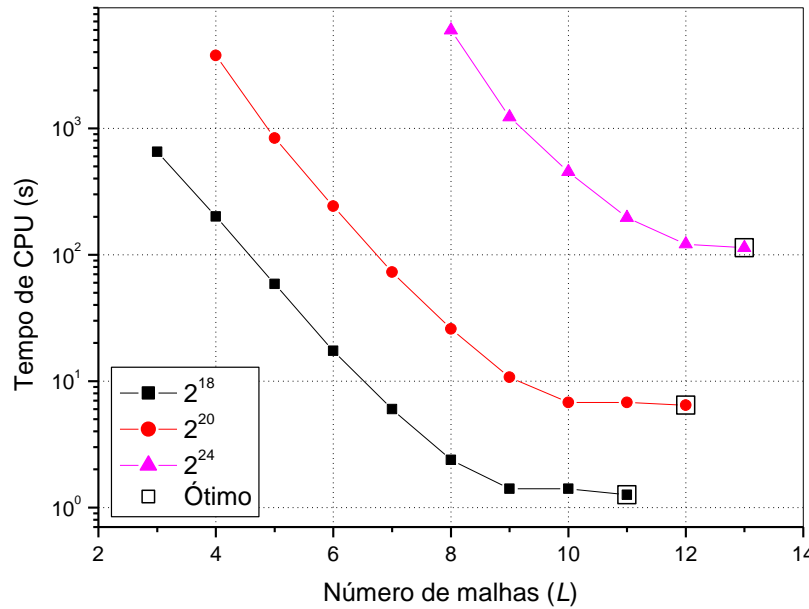


Figura 5.2: Tempo de CPU versus número de níveis para o problema linear em malhas triangulares

Na Fig. 5.2 pode-se observar a influência do número de níveis no tempo de CPU. Nesta figura podem ser observados os resultados obtidos para o número de níveis, no problema de minimização do tempo de CPU, ou seja: para $L_{máximo} = 11$ (que corresponde ao problema com 2^{18} incógnitas), tem-se $L_{ótimo} = 11$, para $L_{máximo} = 12$ (que corresponde ao problema com 2^{20} incógnitas), tem-se $L_{ótimo} = 12$ e para $L_{máximo} = 13$ (que corresponde ao problema com 2^{24} incógnitas) tem-se $L_{ótimo} = 13$. Desta análise pode-se concluir que existe um valor para o número de níveis que resulta no menor tempo de CPU, este valor é $L_{máximo}$, logo $L_{ótimo} = L_{máximo}$. Desta análise conclui-se que o tamanho de problema a ser resolvido não influencia no valor a ser empregado para o número de níveis.

Da Fig. 5.2 também pode-se concluir que, quanto menor o número de níveis empregados na resolução, maior é tempo de CPU necessário para se resolver o AMG. Estes resultados concordam com os obtidos anteriormente para malhas quadrangulares, em que o número ótimo de níveis é o máximo. Concordam também com Gaspar et al. (2009), que apesar de não fazerem um estudo detalhado deste parâmetro, empregam o número máximo de níveis na resolução dos problemas com GMG. Da mesma forma que para malhas

quadrangulares, não foram encontrados estudos a respeito do efeito do número de níveis no tempo de CPU para o AMG em malhas triangulares.

5.1.3 Fator de redução de malha (θ)

Para o estudo deste parâmetro em malhas triangulares é adotada a mesma metodologia empregada para malhas quadrangulares. Os valores ótimos para os parâmetros já conhecidos são empregados, ou seja, $\nu = 1$ e $L = L_{\text{máximo}}$, e o fator de forte dependência na malha grossa é o mesmo aplicado anteriormente, $\varepsilon = 0,35$. Para este parâmetro são testados os mesmos valores já empregados para malhas quadrangulares, ou seja, $\theta = 0,25$, $\theta = 0,20$, $\theta = 0,4$, $\theta = 0,11$ e $\theta = 0,0625$, que são valores que possuem correspondência com o método *multigrid* geométrico, além de serem empregados na literatura.

Para malhas estruturadas triangulares, observou-se que o número de níveis empregado varia conforme o valor testado para θ . Para um determinado tamanho de problema, quando são empregados valores pequenos para θ , como por exemplo $\theta = 0,0625$, observa-se a construção de menos níveis de malha, quando comparado ao problema resolvido com $\theta = 0,4$. Além disso, nota-se a diferença no número de ciclos necessários para se resolver o problema, conforme pode ser observado na Tab. 5.1. Nesta tabela são observados os valores correspondentes ao problema com mais incógnitas, sendo que para os demais tamanhos de problema, os resultados obtidos são análogos. Nos testes computacionais que seguem, foram considerados todos os níveis de malha.

Tabela 5.1: Número de níveis e de ciclos para o problema linear resolvido em malhas triangulares para diferentes valores de θ

θ	Níveis	Iterações externas (Ciclos)
0,0625	12	24
0,11	12	25
0,2	14	33
0,25	13	29
0,4	14	47

Da observação da Tab. 5.1 nota-se que o problema resolvido com $\theta = 0,0625$ emprega menos níveis de malha, além de gastar menos ciclos para atingir a tolerância desejada. Desta forma, pode-se concluir que o valor empregado para θ influencia no número de níveis a ser

construído pelo AMG. Na Fig. 5.3 pode-se observar o comportamento do fator de redução de malha com a variação do número de incógnitas. Para cada tamanho de problema foram obtidos os tempos de CPU para os diferentes valores de θ . Nota-se que a diferença entre os tempos de CPU obtida para cada um dos valores testados é considerável. Nota-se também que existe um valor que resulta no menor tempo de CPU, para a malha mais refinada. Este valor é $\theta = 0,0625$.

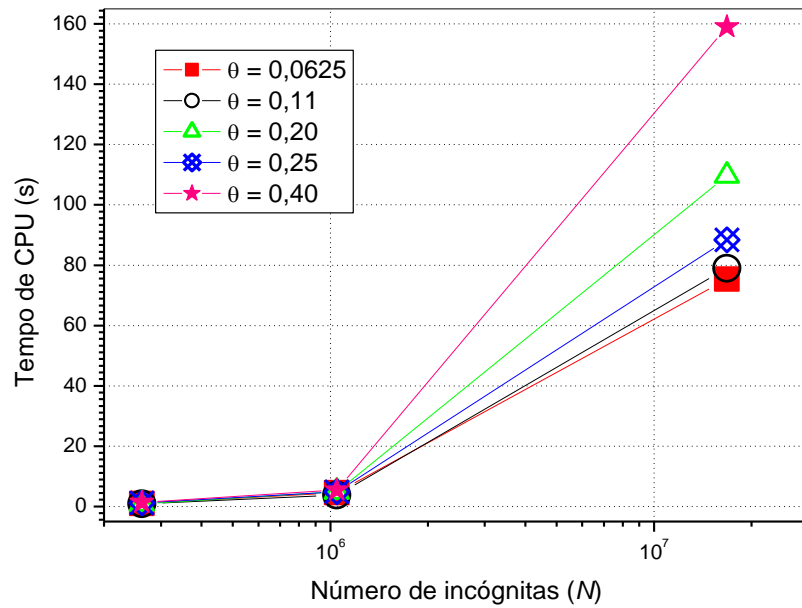


Figura 5.3: Influência do fator de redução de malha no tempo de CPU para o problema linear em malhas triangulares

Da Fig. 5.3 nota-se também que o valor padrão na literatura, $\theta = 0,25$, não apresenta um bom desempenho quando comparado aos demais valores testados, pois o tempo de CPU obtido na resolução da malha com mais incógnitas para este valor, não é o menor. A seguir, é calculada a diferença (em termos percentuais), entre o tempo de CPU usando-se o valor ótimo e cada um dos valores de θ . Estas porcentagens podem ser observadas na Tab. 5.2.

Tabela 5.2: Diferenças entre os tempos de CPU para θ considerando o problema linear em malhas triangulares

θ	262144	1048576	16777216
0,0625	13,9%	9,0%	-----
0,11	0,5%	-----	4,4%
0,2	-----	17,7%	37,9%
0,25	12,6%	16,5%	17,6%
0,4	27,8%	43,1%	100,4%

Esta análise tem o objetivo de encontrar o valor que apresenta o melhor desempenho médio para este parâmetro, levando em consideração que os problemas de maior importância são os problemas com mais incógnitas.

Para o problema em questão observa-se que existe um valor, $\theta = 0,0625$, que apresenta o comportamento esperado, quando observada a Tab. 5.2. Comportamento esperado é aquele em que as porcentagens obtidas diminuem com o aumento do número de incógnitas, visto que o principal objetivo de trabalhos em engenharia é de se resolver problemas nas malhas mais refinadas possíveis. Desta forma, pode-se concluir que o valor a ser empregado nas simulações que seguem é $\theta = 0,0625$. Da Tab. 5.2 também se observa que o valor amplamente empregado na literatura, $\theta = 0,25$, não apresenta um comportamento satisfatório, pois apesar de apresentar uma porcentagem menor na malha com menos incógnitas, quando comparado a $\theta = 0,0625$, as porcentagens calculadas aumentam com o número de incógnitas, para o valor amplamente empregado na literatura.

Foram testados valores menores que $\theta = 0,0625$ para malhas triangulares, mas o resultado não foi satisfatório, pois não houve solução para o problema. Acredita-se que este fato é devido ao valor empregado para θ ser muito pequeno. Desta forma, a malha imediatamente mais grossa tem mais pontos, praticamente coincidindo com a malha mais refinada. Para este parâmetro, em malhas triangulares, acredita-se que o menor valor testado, $\theta = 0,0625$ seja o valor limite para este parâmetro, quando são considerados valores que possuem correspondência com as razões de engrossamento do GMG.

5.1.4 Fator de forte dependência na malha grossa (ε)

Para o estudo deste parâmetro são considerados os valores ótimos encontrados nos estudos anteriores, ou seja, para o número de iterações internas, $\nu = 1$, para o número de níveis de malha, $L = L_{\text{máximo}}$ e para o fator de redução de malha, $\theta = 0,0625$. A metodologia empregada no estudo deste parâmetro é a mesma já aplicada para malhas estruturadas quadrangulares. Os valores testados são: $\varepsilon = 0,20$, $\varepsilon = 0,25$, $\varepsilon = 0,30$, $\varepsilon = 0,35$, $\varepsilon = 0,40$, $\varepsilon = 0,45$ e $\varepsilon = 0,50$. Durante a revisão bibliográfica não foram encontrados estudos a respeito da influência deste parâmetro no tempo de CPU.

Na Fig. 5.4 pode ser observado o comportamento dos vários valores testados para ε , para os diversos números de incógnitas. Pela análise desta figura não é possível concluir qual

valor apresenta o melhor desempenho. Para todos os valores testados de ε , nota-se que os tempos de CPU estão muito próximos. Para que se possa concluir qual valor a ser empregado nas próximas simulações, é feito o estudo das porcentagens do ganho no tempo de CPU, para os diferentes valores de ε , para os três tamanhos de problema aqui resolvidos. Esta análise pode ser observada na Tab. 5.3.

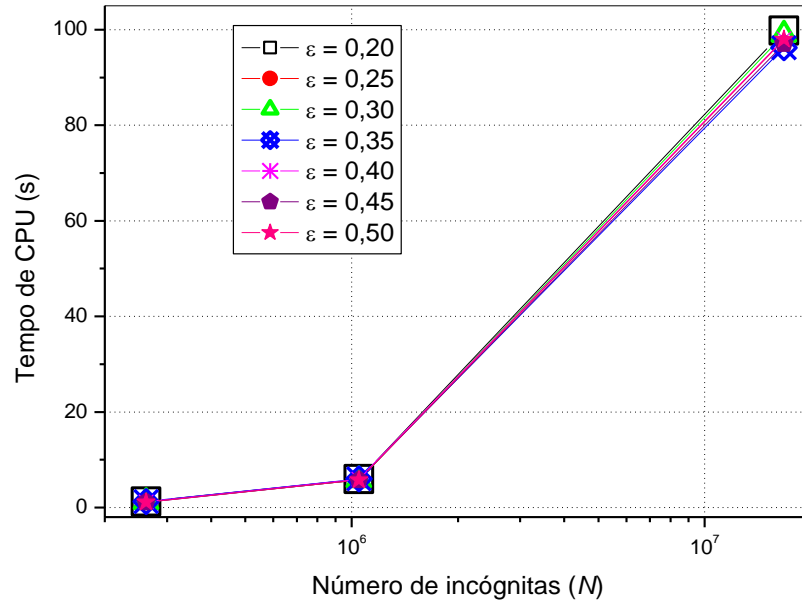


Figura 5.4: Efeito do fator de forte dependência na malha grossa no tempo de CPU para o problema linear em malhas triangulares

Tabela 5.3: Diferenças entre os tempos de CPU para ε considerando o problema linear em malhas triangulares

ε	262144	1048576	16777216
0,20	6,7%	3,3%	3,7%
0,25	0,9%	-----	1,9%
0,30	0,4%	0,1%	2,9%
0,35	9,9%	5,3%	-----
0,40	1,3%	3,4%	2,0%
0,45	-----	0,3%	0,8%
0,50	-----	0,3%	1,8%

Da análise desta tabela, nota-se que o valor a ser empregado nas próximas simulações é $\varepsilon = 0,35$. Para este valor, a porcentagem que expressa a diferença entre os valores aplicados e o ótimo diminui com o refino da malha, além de ser o valor ótimo para a malha mais refinada. Este é o valor mais empregado na literatura, considerado padrão. Desta forma, o valor a ser empregado nas próximas simulações é $\varepsilon = 0,35$. Este valor também coincide com o encontrado para a equação de Laplace em malhas quadrangulares. Nota-se que, $\varepsilon = 0,45$

apesar de ser o valor ótimo para a malha com menos incógnitas apresenta os valores para as porcentagens aumentando com o número de incógnitas.

5.1.5 Número de incógnitas (N)

Com os valores ótimos obtidos para cada um dos parâmetros estudados nas seções anteriores, é feito o estudo do efeito do número de incógnitas no tempo de CPU. O problema linear é resolvido em diversos tamanhos de malha (variando de 4 até 16777216 incógnitas). Os resultados obtidos para o AMG são comparados aos resultados obtidos para o SG. Esta comparação pode ser observada na Fig. 5.5, onde nota-se o comportamento dos tempos de CPU obtidos para o AMG e para o SG, com o aumento do número de incógnitas. Este comportamento é o esperado, visto que, a curva que representa o AMG está afastada da curva do SG, além de apresentarem inclinações diferentes.

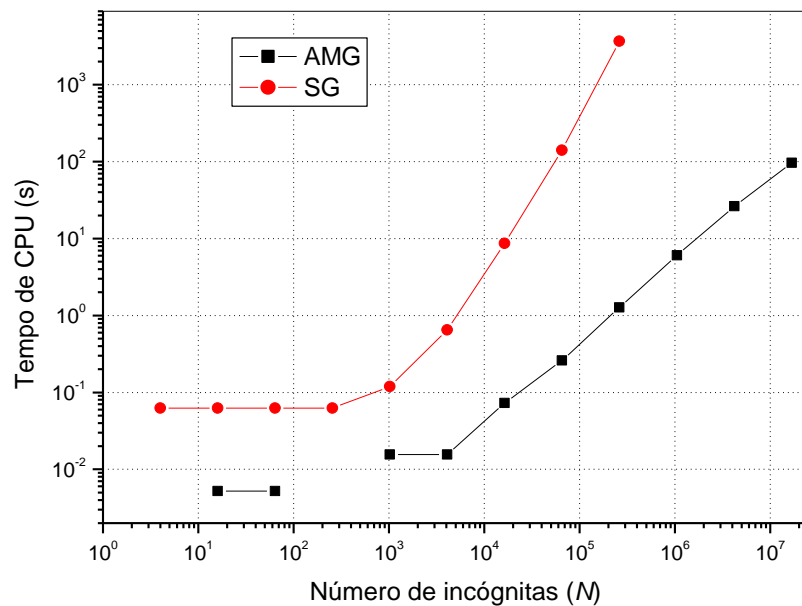


Figura 5.5: Efeito do número de incógnitas no tempo de CPU para o problema linear em malhas triangulares

Um estudo a respeito da inclinação das curvas é feito com o uso do ajuste de curvas pelo método dos mínimos quadrados. Em particular, o ajuste é feito por uma curva geométrica, conforme Eq. (3.4). Na Tab. 5.4 podem ser observados os valores para o expoente p e o coeficiente c relativos a esta equação, para os dois métodos empregados. O método *multigrid* ideal é aquele em que o expoente p está próximo da unidade.

Tabela 5.4: Valores de p e c para o problema linear em malhas triangulares

Método empregado	Expoente p	Coefficiente c
AMG	1,05913	2,34300E-06
SG	1,87915	2,26339E-08

Nota-se que os resultados obtidos para a equação de Laplace discretizada em malhas triangulares são os esperados. O expoente p para o método *multigrid* algébrico está próximo da unidade enquanto o expoente para o *singlegrid* está mais próximo de 2 do que de 1. Nota-se também que o coeficiente c ilustra o distanciamento entre as curvas da Fig. 5.5.

5.1.6 Otimização do algoritmo para o AMG

Através do estudo acima descrito, são encontrados parâmetros ótimos para o problema linear discretizado em malhas triangulares. Desta forma, chega-se a um algoritmo com parâmetros ótimos, que é aplicado à resolução deste problema. Com esta otimização obtém-se um ganho em termos de tempo de CPU. Os valores para os parâmetros empregados nos algoritmos podem ser visualizados na Tab. 5.5. A comparação para os tempos de CPU obtidos para o algoritmo padrão e o otimizado podem ser observados na Fig. 5.6, onde nota-se que a diferença entre os tempos aumenta com o número de incógnitas.

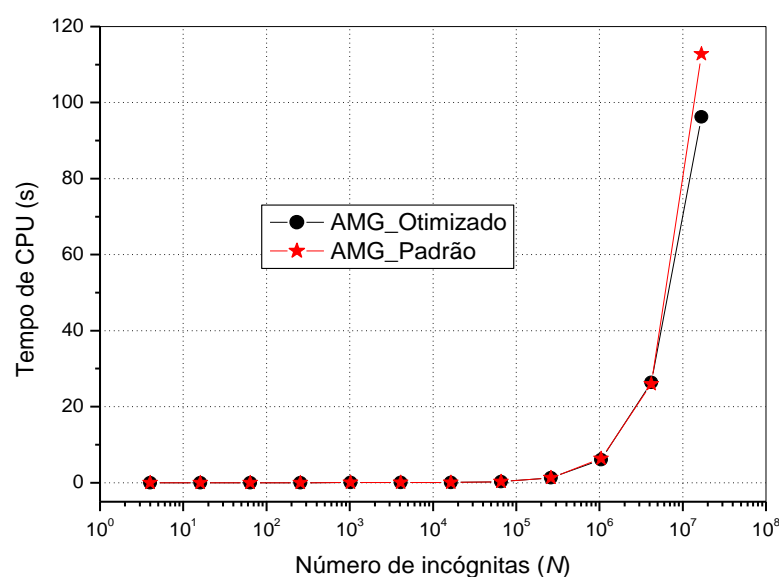


Figura 5.6: AMG otimizado versus AMG padrão para o problema linear em malhas triangulares

Tabela 5.5: Parâmetros estudados para o AMG para o problema linear em malhas triangulares

Parâmetro estudado	AMG padrão	AMG otimizado
Número de iterações internas (ν)	1	1
Número de níveis (L)	$L_{\text{máximo}}$	$L_{\text{máximo}}$
Fator de redução de malha (θ)	0,25	0,0625
Fator de forte dependência na malha grossa (ε)	0,35	0,35

Da Tab. 5.5 pode-se concluir que apenas um dos parâmetros em estudo teve seu valor modificado em relação ao algoritmo padrão. Com esta modificação, o ganho em tempo de CPU é de 17,2% para a malha mais refinada.

Outra análise a ser feita com base nestes resultados é o estudo dos expoentes p e dos coeficientes c , da Eq. (3.4), para o AMG padrão e otimizado. Além destas constantes, pode ser comparado o número de ciclos (iteraões externas) que o AMG leva para atingir a tolerância estimada em 10^{-8} . Para o resultado do número de ciclos, é considerada a malha mais refinada. Estas comparações entre os dois algoritmos podem ser observadas na Tab. 5.6.

Tabela 5.6: Valores de p e c para AMG padrão e otimizado para o problema linear em malhas triangulares

Método empregado	Expoente p	Coefficiente c	Ciclos
AMG Padrão	1,07737	2,95447E-06	29
AMG Otimizado	1,05913	2,34300E-06	25

Da Tab. 5.6 nota-se que o expoente p do algoritmo otimizado está mais próximo da unidade que o valor obtido para o algoritmo padrão. Nota-se também que os coeficientes c estão muito próximos para os dois algoritmos. Observa-se ainda que o número de iterações externas (ciclos) empregadas para se atingir a tolerância desejada é menor para o algoritmo otimizado.

5.2 Equação de Laplace – Problema senoidal

Nesta seção são descritos os resultados obtidos para a equação de Laplace resolvida em um domínio de cálculo quadrado, com condições de contorno dadas por uma função senoidal ao norte. A discretização do domínio de cálculo é feita com malhas triangulares com o uso do método dos volumes finitos. O objetivo deste estudo é obter parâmetros ótimos que, quando aplicados ao algoritmo, resultem no menor tempo de CPU. Os resultados aqui obtidos

são comparados aos já existentes para o problema linear, resolvido em malhas triangulares (MVF), e também para a equação de Laplace em malhas quadrangulares (MDF). Os parâmetros analisados são: número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ), fator de forte dependência na malha grossa (ε). Depois de consideradas as otimizações são analisados o efeito do número de incógnitas no tempo de CPU. São comparados também o tempo de CPU, o expoente p e o coeficiente c da Eq. (3.4), e o número de iterações externas empregado para os algoritmos padrão e otimizado.

5.2.1 Número de iterações internas (ν)

De acordo com os testes realizados anteriormente, este parâmetro tem seu valor variado entre 1 e 20 para os problemas pequeno, médio e grande, conforme descrito na seção anterior. Para a realização destes testes computacionais são considerados todos os níveis de malha, $L = L_{máximo}$, fator de redução de malha $\theta = 0,25$ e fator de forte dependência na malha grossa $\varepsilon = 0,35$, conforme empregado na literatura. Na Fig. 5.7 podem ser observados os resultados obtidos para este estudo. Nota-se que existe um número de iterações internas que resulta no menor tempo de CPU, independente do tamanho de problema que está sendo resolvido.

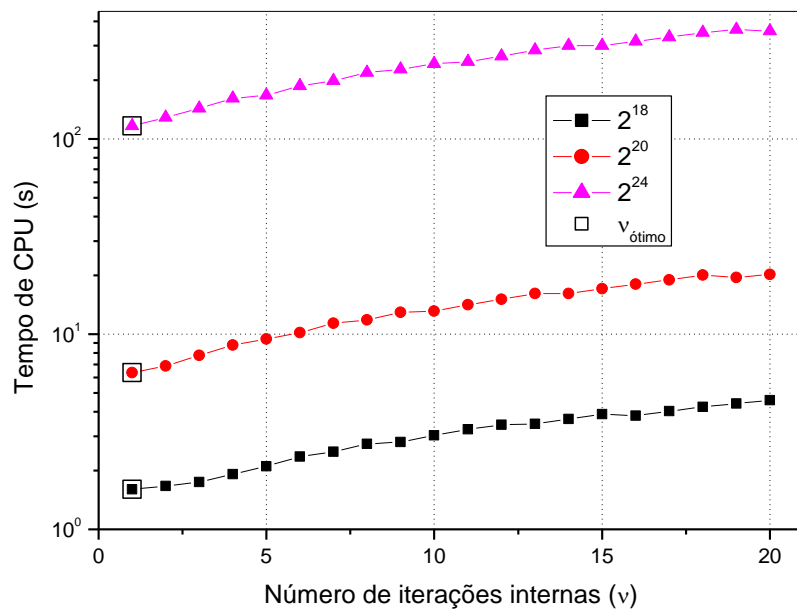


Figura 5.7: Tempo de CPU *versus* número de iterações internas para o problema senoidal em malhas triangulares

Para o problema senoidal discretizado em malhas triangulares, o número ótimo de iterações internas é 1, ou seja, $\nu_{\text{ótimo}} = 1$. Observa-se também que, quanto maior o número de iterações utilizadas, maior é o tempo de CPU gasto para se resolver o problema. No trabalho de Gaspar et al. (2009) a conclusão é de que $\nu = 1$ é o valor ótimo para a resolução dos problemas. Para o problema em questão, observou-se que o valor de $\nu_{\text{ótimo}}$ é o mesmo encontrado para o problema linear, quando ambos são resolvidos em malhas triangulares. O valor aqui encontrado difere do valor ótimo encontrado para malhas quadrangulares. Com base na discussão acima, o valor a ser empregado nas próximas simulações é o valor ótimo, ou seja, $\nu_{\text{ótimo}} = 1$.

5.2.2 Número de níveis (L)

Para o estudo do efeito do número de níveis no tempo de CPU com a variação do número de incógnitas, são considerados: número ótimo de iterações internas, $\nu = 1$, fator de redução de malha padrão, $\theta = 0,25$ e fator de forte dependência na malha grossa padrão, $\varepsilon = 0,35$.

Os valores empregados para o número de níveis são variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia com o tamanho do problema, ou seja, $L_{\text{máximo}}, L_{\text{máximo}} - 1, L_{\text{máximo}} - 2$, e assim sucessivamente. Para a determinação deste valor para o parâmetro em questão, é observado o crescimento do tempo de CPU, com a diminuição do número de níveis. O valor para L tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele cuja resolução do problema leva em torno de 3 horas. Deve-se observar que com este valor adotado já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar. Se fossem feitos testes com outros valores para L (menores que o mínimo aqui adotado), os tempos de CPU resultantes estariam entre 20 e 24 horas, valores que dificultariam a visualização gráfica deste parâmetro. O comportamento deste parâmetro para os três tamanhos de problema analisados neste trabalho podem ser observados na Fig. 5.8.

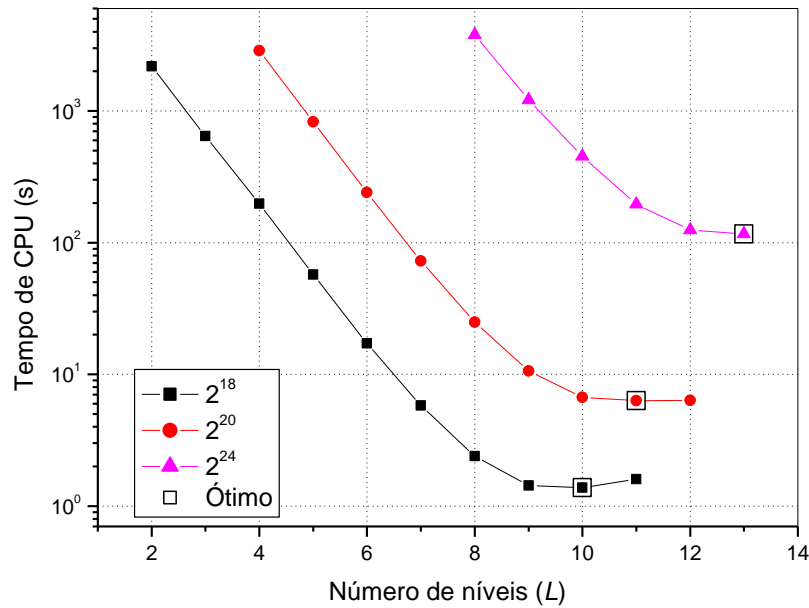


Figura 5.8: Tempo de CPU *versus* número de níveis para o problema senoidal em malhas triangulares

Da observação desta figura, tem-se que, para um determinado tamanho de problema, é variado o número de níveis para a resolução do problema. Desta análise conclui-se que: para $L_{\text{máximo}} = 11$ (que corresponde ao problema com 2^{18} incógnitas), tem-se $L_{\text{ótimo}} = 10$, para $L_{\text{máximo}} = 12$ (que corresponde ao problema com 2^{20} incógnitas), tem-se $L_{\text{ótimo}} = 11$, e para $L_{\text{máximo}} = 13$ (que corresponde ao problema com 2^{24} incógnitas) tem-se $L_{\text{ótimo}} = 13$. Desta análise, pode-se concluir que, para os problemas considerados pequeno e médio, o valor ótimo para o número de níveis é $L_{\text{máximo}} - 1$. Para o problema considerado grande, o valor que resulta no menor tempo de CPU é $L_{\text{máximo}}$. A análise da Fig. 5.8 não é suficiente para se determinar um valor geral para uso posterior. Por este motivo, são calculadas as porcentagens nos ganhos no tempo de CPU, conforme Tab. 5.7.

Tabela 5.7: Diferenças entre os tempos de CPU para L considerando o problema senoidal em malhas triangulares

L	262144	1048576	16777216
$L_{\text{máximo}}$	16,2%	0,9%	-----
$L_{\text{máximo}} - 1$	-----	-----	6,9%

Da análise da Tab. 5.7 conclui-se que o número de níveis que obteve o melhor desempenho médio é o número máximo ($L = L_{\text{máximo}}$), pois, as porcentagens com relação aos valores ótimos diminuem com o aumento do número de incógnitas. O melhor desempenho

médio leva em conta os problemas com mais incógnitas, que são de interesse prático. Desta forma, o valor a ser empregado nas próximas simulações é $L_{máximo}$.

Da Fig. 5.8, observa-se que, quanto menor o número de níveis de malha empregados, maior é o tempo de CPU gasto para se resolver o problema. Estes resultados concordam com os descritos anteriormente para a equação de Laplace discretizada em malhas triangulares (problema linear) e malhas quadrangulares (problema linear e senoidal). Concordam ainda com o trabalho de Gaspar et al. (2009), que utiliza todas as malhas para se resolver um problema em malhas triangulares com o uso do GMG.

5.2.3 Fator de redução de malha (θ)

Para se avaliar o efeito do fator de redução de malha no tempo de CPU são considerados os parâmetros ótimos já encontrados anteriormente. Para o número de iterações internas, $\nu = 1$ e, para o número de malhas, $L = L_{máximo}$. Além disso, para o fator de forte dependência na malha grossa é empregado $\varepsilon = 0,35$. Os valores a serem testados para este parâmetro são os mesmos já empregados para o problema linear, quando foi feita esta mesma análise. Conforme observado para o problema linear, na resolução do problema senoidal também é observada a variação do número máximo de níveis conforme o valor empregado para θ . Esta variação pode ser observada na Tab. 5.8, onde pode-se observar também o número de iterações externas (ciclos) empregados na resolução do problema, para a malha com mais incógnitas, sendo que os resultados obtidos para as malhas com menos incógnitas são análogos.

Tabela 5.8: Número de níveis e de ciclos para o problema senoidal resolvido em malhas triangulares para diferentes valores de θ

θ	Níveis	Iterações externas (Ciclos)
0,0625	12	25
0,11	12	26
0,2	14	36
0,25	13	30
0,4	14	51

Da observação da Tab. 5.8 nota-se que o problema resolvido com $\theta = 0,0625$ emprega menos níveis de malha, além de gastar menos ciclos para atingir a tolerância desejada. Desta forma, pode-se concluir que o valor empregado para θ influencia no número de níveis a ser

construído pelo AMG. O comportamento para o fator de redução de malha com a variação do número de incógnitas para o problema senoidal pode ser observado na Fig. 5.9.

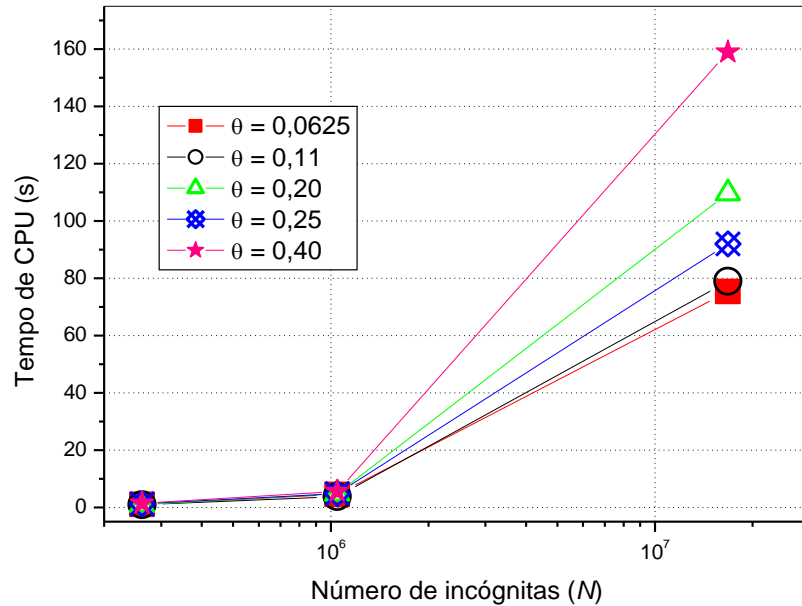


Figura 5.9: Influência do fator de redução de malha no tempo de CPU para o problema senoidal em malhas triangulares

Da observação desta figura nota-se que, o valor empregado para θ , influencia no tempo de CPU. Para a malha mais refinada pode-se observar mais claramente a diferença entre os tempos obtidos para os diferentes valores de θ . De acordo com a figura, o valor que apresentou o melhor desempenho foi $\theta = 0,0625$. O valor considerado padrão pela literatura, $\theta = 0,25$, não apresentou o melhor desempenho neste estudo. Conforme metodologia aplicada para o problema linear, resolvido anteriormente, foram calculadas as porcentagens dos ganhos de tempo de CPU, conforme Tab. 5.9.

Tabela 5.9: Diferenças entre os tempos de CPU para θ considerando o problema senoidal em malhas triangulares

θ	262144	1048576	16777216
0,0625	12%	4,1%	-----
0,11	4,2%	-----	4,5%
0,2	----	29,8%	45,6%
0,25	42,6%	18,6%	17,5%
0,4	58,1%	47,1%	111,1%

Da análise da Tab. 5.9 pode-se concluir que o valor para θ que apresenta o melhor desempenho médio para a resolução do problema senoidal é $\theta = 0,0625$. Para este valor,

nota-se que as porcentagens diminuem com o refino da malha, além de ser o valor ótimo para a malha mais refinada. Desta forma, para as próximas simulações é utilizado $\theta = 0,0625$. Este valor coincide com o encontrado para o problema linear, quando resolvido em malhas triangulares, mas difere do valor encontrado para malhas quadrangulares.

5.2.4 Fator de forte dependência na malha grossa (ε)

Para avaliar o efeito do fator de forte dependência na malha grossa no tempo de CPU são considerados os parâmetros que apresentaram o melhor desempenho médio nas análises anteriores: para o número de iterações internas $\nu = 1$, para o número de malhas, $L = L_{\text{máximo}}$ e para o fator de redução de malha, $\theta = 0,0625$. A mesma metodologia aplicada nos estudos deste parâmetro para as outras equações é considerada nesta análise. Os resultados obtidos para o fator de forte dependência na malha grossa com a variação do tamanho do problema podem ser observados na Fig. 5.10.

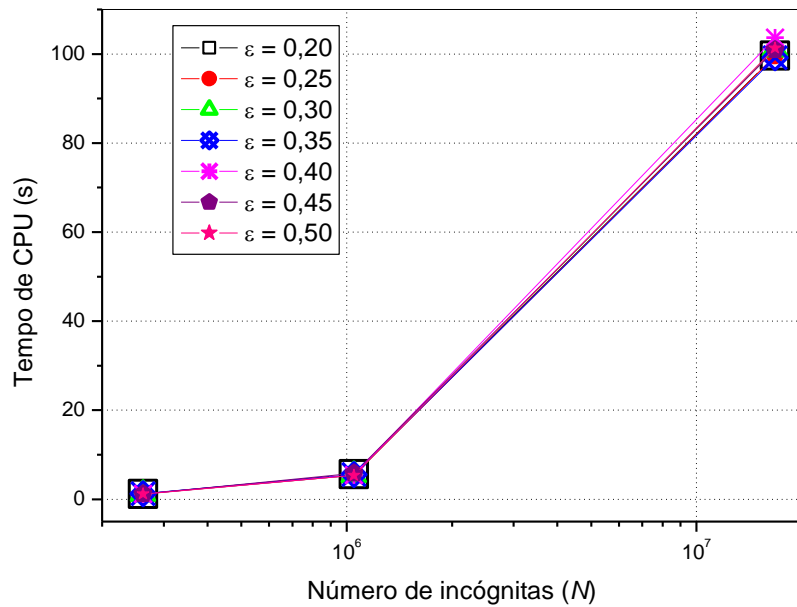


Figura 5.10: Influência do fator de forte dependência na malha grossa no tempo de CPU para o problema senoidal em malhas triangulares

Desta figura pode-se concluir que os tempos de CPU obtidos para os valores testados estão muito próximos. A observação da Fig. 5.10 não é suficiente para determinar qual o valor que apresenta o melhor desempenho médio. Por este motivo, para a determinação deste valor,

são obtidas as porcentagens do ganho em tempo de CPU para cada um dos valores estudados, conforme Tab. 5.10.

Tabela 5.10: Diferenças entre os tempos de CPU para ε considerando o problema senoidal em malhas triangulares

ε	262144	1048576	16777216
0,20	1,7%	4,0%	0,4%
0,25	0,4%	0,6%	0,4%
0,30	-----	0,8%	1,8%
0,35	5,2%	3,4%	-----
0,40	-----	0,3%	4,5%
0,45	1,3%	8,8%	1,8%
0,50	1,3%	-----	2,2%

Da análise desta tabela, nota-se que o valor a ser empregado nos próximos testes computacionais é $\varepsilon = 0,35$. Para este valor, a porcentagem que expressa a diferença entre os valores aplicados e o ótimo diminui com o refino da malha, além de ser o valor ótimo para a malha mais refinada. Este valor coincide com o encontrado para o problema linear resolvido em malhas triangulares, e para os problemas linear e senoidal resolvidos em malhas quadrangulares.

5.2.5 Número de incógnitas (N)

Para avaliar o efeito do número de incógnitas sobre o tempo de CPU são considerados todos os parâmetros que apresentaram o melhor desempenho médio, obtidos nas análises anteriores. Os parâmetros são: número de iterações internas $\nu=1$, número de malhas $L = L_{\text{máximo}}$, fator de redução de malha $\theta=0,0625$ e fator de forte dependência na malha grossa $\varepsilon=0,35$. Para estes testes são resolvidos problemas onde o número de incógnitas varia de 4 até 16777216 incógnitas, considerando os volumes reais.

Os resultados obtidos para o AMG são comparados aos obtidos para SG, conforme pode ser observado na Fig. 5.11. O comportamento do tempo de CPU com o aumento do número de incógnitas é observado nesta figura. Nota-se que é um comportamento muito próximo do apresentado na Fig. 5.5, onde esta mesma análise é feita para o problema linear. Nota-se que o AMG e o SG possuem inclinações diferentes. Percebe-se ainda que o AMG é mais rápido que o SG.

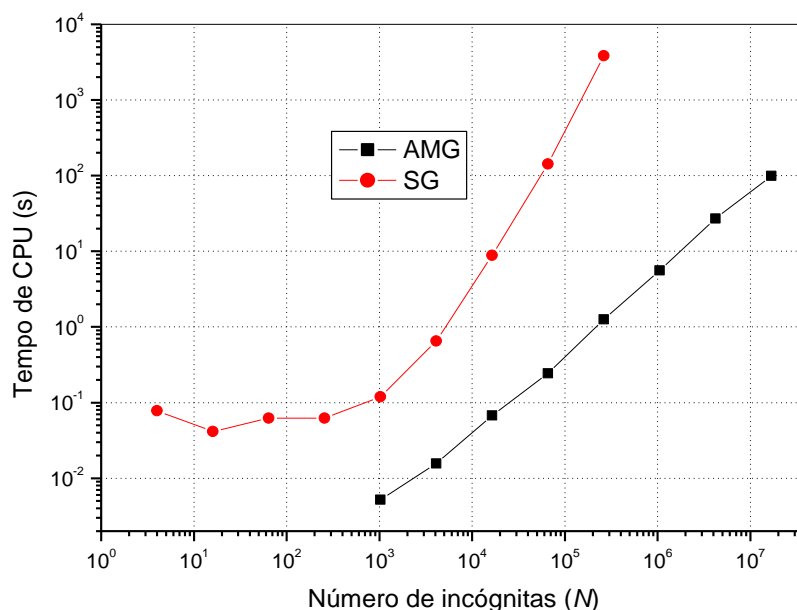


Figura 5.11: Efeito do número de incógnitas no tempo de CPU para o problema senoidal em malhas triangulares

Um estudo pode ser feito a respeito da inclinação das curvas apresentadas na Fig. 5.11, com o uso do ajuste de curvas pelo método dos mínimos quadrados. Este ajuste é feito por uma curva geométrica, conforme a Eq. (3.4). Os resultados obtidos para esta análise podem ser observados na Tab. 5.11, que traz os valores para os expoentes p e os coeficientes c da Eq. (3.4) dos dois métodos aqui empregados para a resolução do problema em questão.

Tabela 5.11: Valores de p e c para o problema senoidal em malhas triangulares

Método empregado	Expoente p	Coefficiente c
AMG	1,04033	2,99400E-06
SG	1,88590	8,96788E-08

Pela Tab. 5.11 pode-se concluir que os expoentes p obtidos para as curvas ilustradas na Fig. 5.11 são os esperados. O expoente obtido para o AMG está próximo da unidade, enquanto que para o SG, este valor está mais próximo de 2 do que de 1, conforme o esperado. Nota-se que os valores obtidos para o coeficiente c , que refletem o distanciamento entre as curvas, não estão próximos entre si. Estes resultados concordam com os apresentados pelas curvas da Fig. 5.11. Também estão de acordo com os obtidos para o problema linear resolvido em malhas triangulares e também com os resultados obtidos para malhas quadrangulares.

5.2.6 Otimização do algoritmo para o AMG

Nesta seção são comparados os resultados obtidos para o algoritmo otimizado (com os parâmetros ótimos definidos nas seções anteriores) e para o algoritmo padrão, ambos para o problema senoidal. Para tanto, são comparados os tempos de CPU para os dois algoritmos, e também o número de iterações externas (ciclos) de cada algoritmo para se atingir a tolerância estipulada. Tanto o tempo de CPU quanto o número de ciclos tiveram seus valores reduzidos com a otimização. Os valores empregados para os parâmetros dos algoritmos padrão e otimizado são dados pela Tab. 5.12.

Tabela 5.12: Parâmetros estudados para o AMG para o problema senoidal em malhas triangulares

Parâmetro estudado	AMG padrão	AMG otimizado
Número de iterações internas (ν)	1	1
Número de níveis (L)	$L_{\text{máximo}}$	$L_{\text{máximo}}$
Fator de redução de malha (θ)	0,25	0,0625
Fator de forte dependência na malha grossa (ε)	0,35	0,35

Da observação da Tab. 5.12 pode-se notar que apenas um dos parâmetros em estudo apresenta um valor ótimo diferente do aplicado no algoritmo padrão. Apesar desta única modificação no algoritmo, o ganho em tempo de CPU é de 16% na malha mais refinada. A comparação para os tempos de CPU entre o algoritmo padrão e o otimizado pode ser observada na Fig. 5.12, onde nota-se que quanto maior o número de incógnitas, maior é a diferença entre os tempos de CPU.

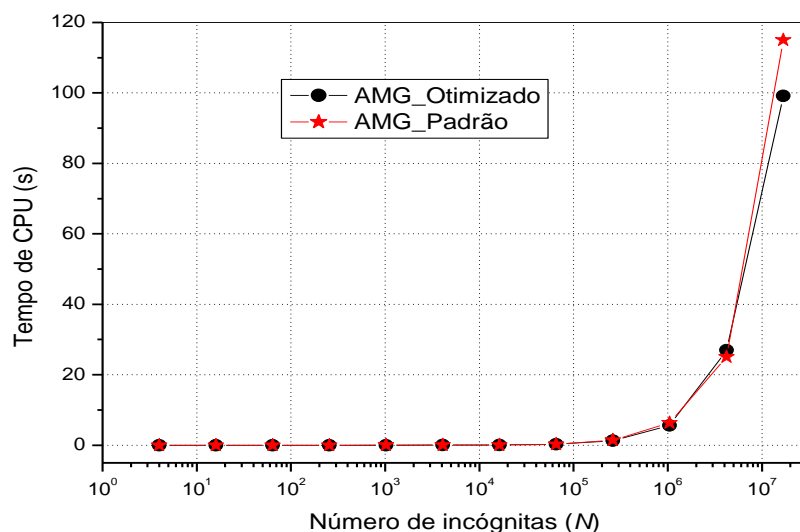


Figura 5.12: AMG otimizado versus AMG padrão para o problema senoidal em malhas triangulares

Para complementar este estudo são calculados os expoentes p e os coeficientes c da Eq. (3.4) para o AMG padrão e o otimizado. Outra comparação a ser feita entre os dois algoritmos é para o número de ciclos (iterações externas) que são realizados para se atingir a tolerância desejada, 10^{-8} . Estas comparações são dadas na Tab. 5.13.

Tabela 5.13: Valores de p e c para AMG padrão e otimizado para o problema senoidal em malhas triangulares

Método empregado	Expoente p	Coeficiente c	Ciclos
AMG Padrão	1,07357	2,02806E-06	30
AMG Otimizado	1,04033	2,99400E-06	25

O desempenho apresentado pelo AMG otimizado é superior ao padrão, como pode ser observado na Tab. 5.13. O expoente p , está mais próximo da unidade para o algoritmo otimizado. Nota-se também que os coeficientes c estão próximos entre si. Nesta mesma tabela observa-se que o número de iterações externas é reduzido. Este resultado condiz com os obtidos anteriormente, pois se obtêm uma redução tanto para o expoente p quanto para o número de ciclos, para os problemas resolvidos em malhas triangulares.

5.3 Comparativo para malhas triangulares

Neste capítulo, são estudados dois problemas partindo da equação de Laplace em malhas triangulares, discretizadas com o uso do método dos volumes finitos: problema linear e problema senoidal. O objetivo desta seção é comparar os resultados obtidos para cada um dos parâmetros estudados, nos dois problemas em questão. São analisados os quatro parâmetros estudados: número de iterações internas (ν), número de níveis de malha (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). Além disso, são comparados os resultados obtidos para o tempo de CPU, expoente p e o coeficiente c da Eq. (3.4) e também o número de iterações externas empregadas para se atingir a tolerância desejada. Para estas análises é utilizada a malha mais refinada, pois para os outros tamanhos de problema, os resultados obtidos são análogos.

5.3.1 Número de iterações internas (ν)

O comparativo para o número de iterações internas, para as duas situações abordadas a partir da equação de Laplace, é observado na Fig. 5.13. Com esta análise, pode-se concluir que existe uma tendência para o número ótimo de iterações internas, quando a equação de Laplace é resolvida em malhas triangulares, ou seja, $\nu_{\text{ótimo}} = 1$.

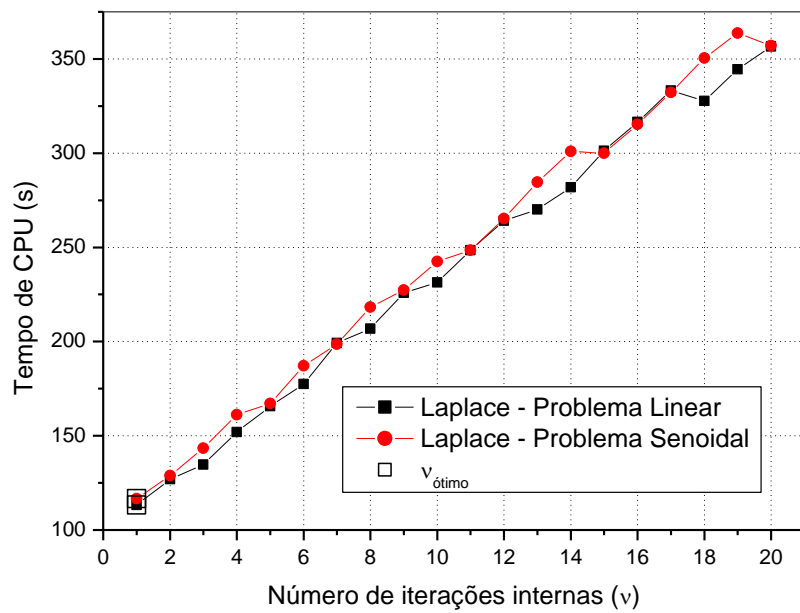


Figura 5.13: Número de iterações internas para malhas triangulares

5.3.2 Número de níveis (L)

Um estudo comparativo para o número ótimo de níveis entre os problemas que são abordados em malhas triangulares pode ser observado na Fig. 5.14. Da análise desta figura pode-se concluir que, para malhas triangulares o ideal é se utilizar todos os níveis de malha possível, ou seja, $L = L_{\text{máximo}}$.

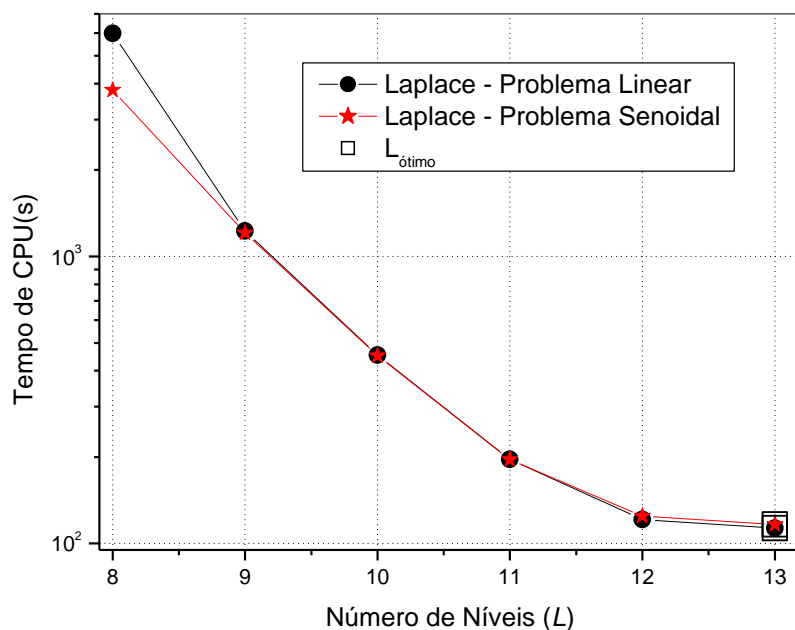


Figura 5.14: Número de níveis para malhas triangulares

5.3.3 Fator de redução de malha (θ)

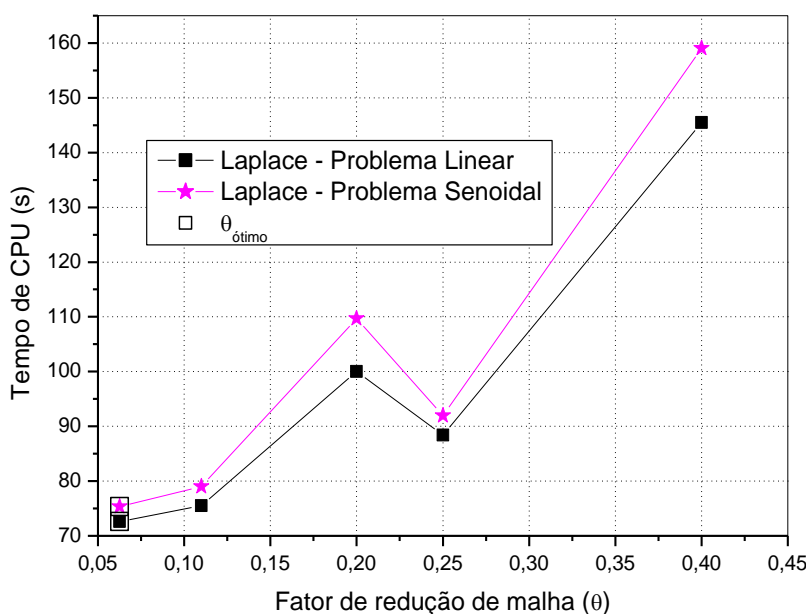


Figura 5.15: Fator de redução de malha para malhas triangulares

Os valores encontrados para o fator de redução de malha em cada um dos problemas em estudo também são comparados. Na Fig. 5.15 podem ser observados os valores obtidos para a malha mais refinada. Da observação desta figura pode-se concluir que o valor amplamente empregado pela literatura não é o valor que apresenta o melhor desempenho para as equações resolvidas em malhas triangulares. O valor do fator de redução de malha faz

diferença para a construção das malhas triangulares, ou seja, quanto menor, melhor para os casos aqui analisados. Desta forma, pode-se concluir que existe uma tendência para o valor a ser empregado para a equação de Laplace discretizada em malhas triangulares. O valor a ser empregado é $\theta = 0,0625$.

5.3.4 Fator de forte dependência na malha grossa (ε)

São comparados os valores encontrados para o fator de forte dependência na malha grossa, considerando os resultados encontrados para a malha mais refinada. Estes resultados podem ser observados na Fig. 5.16. O valor que resulta no menor tempo de CPU, $\varepsilon = 0,35$, coincide com o valor amplamente empregado na literatura. Desta forma, pode-se concluir que existe uma tendência para o valor a ser empregado para a equação de Laplace discretizada em malhas triangulares.

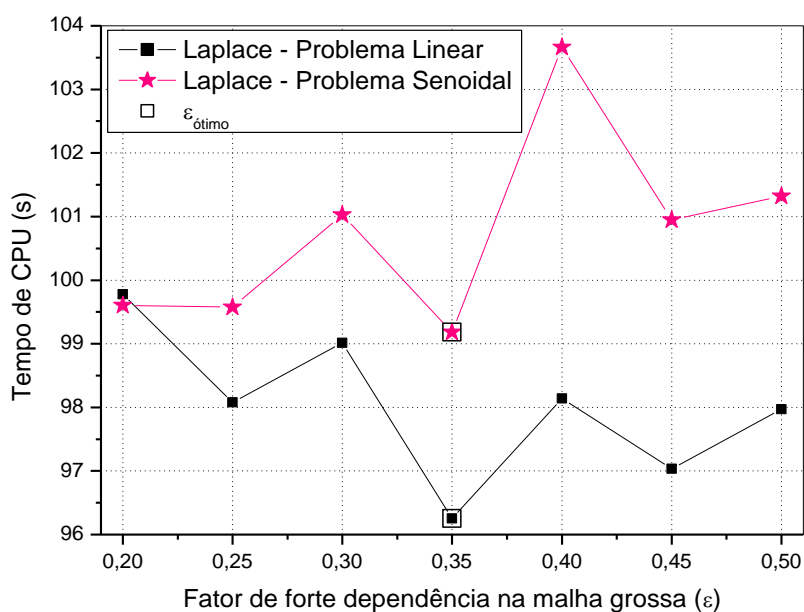


Figura 5.16: Fator de forte dependência na malha grossa para malhas triangulares

5.3.5 Número de incógnitas (N)

Na análise do efeito do número de incógnitas no tempo de CPU, é observado um comportamento padrão para a equação de Laplace, independente do problema que está sendo abordado. Em ambos os casos, o tempo de CPU aumenta quase linearmente com o aumento

do número de incógnitas, quando se utiliza o AMG. Para o SG, as curvas se mantiveram afastadas do AMG, além de apresentarem inclinação maior, conforme pode ser observado na Fig. 5.17, que ilustra as curvas citadas. Um comparativo para os expoentes p e os coeficientes c da Eq. (3.4) para a equação de Laplace pode ser observado na Tab. 5.14.

Tabela 5.14: Valores de p e c para malhas triangulares

Problema	Linear	Senoidal
AMG (c)	2,34300E-06	2,99400E-06
SG (c)	2,26339E-08	8,96788E-08
AMG (p)	1,05913	1,04033
SG (p)	1,87915	1,88590

Nota-se que os expoentes p obtidos tanto para o AMG quanto para o SG estão próximos quando são comparados os tipos de problemas. Mesma observação pode ser feita com relação aos coeficientes c , que também estão próximos para os problemas linear e senoidal.

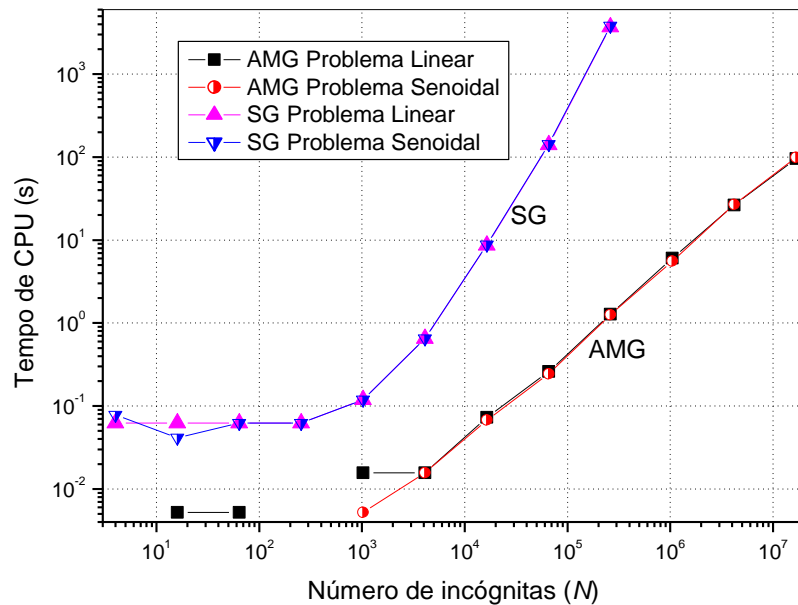


Figura 5.17: Tempo de CPU para AMG, GMG e SG para malhas triangulares

5.3.6 Otimização do algoritmo para o AMG

Com base nos estudos realizados sobre o efeito da variação dos parâmetros do AMG no tempo de CPU, pode-se concluir que, para cada um dos casos analisados, tem-se um conjunto de parâmetros ótimos, que resulta no menor tempo de CPU para a resolução dos

problemas. Com a aplicação destes parâmetros ótimos é obtido um ganho em termos de tempo de CPU, representado pela coluna “Tempo”, na Tab. 5.15. A diferença para o número de iterações externas obtidas entre o algoritmo padrão e o otimizado também pode ser observado na coluna “Ciclos” da mesma tabela. Os resultados apresentados nesta tabela são referentes à malha mais refinada.

Tabela 5.15: Ganhos obtidos com otimizações em malhas triangulares

Problema	Tempo	Ciclos
Linear	17,2%	4
Senoidal	16,0%	5

5.4 Conclusões do Capítulo 5

Neste capítulo foram apresentados os resultados obtidos para a equação de Laplace, resolvida com duas condições de contorno diferentes, discretizada em malhas triangulares: uma considera um problema linear e a outra, um problema senoidal. São obtidos os valores ótimos para número de iterações internas (ν), número de níveis (L), fator de redução de malha (θ) e fator de forte dependência na malha grossa (ε). É avaliado também o efeito do número de incógnitas (N) no tempo de CPU, o expoente p e o coeficiente c da Eq. (3.4) dos métodos envolvidos na resolução das equações e o número de ciclos (iteraões externas). Comparações entre o algoritmo padrão e o otimizado também fazem parte deste trabalho. Destas análises, pode-se concluir que:

- O número de iterações internas (ν) afeta significativamente o tempo de CPU do AMG. O número ótimo é 1, ou seja $\nu_{\text{ótimo}} = 1$, que está de acordo com a literatura.
- O número de níveis (L) empregados para a resolução do AMG afeta significativamente o tempo de CPU. Observou-se que o número de níveis que apresentou o melhor desempenho médio é o valor máximo, ou seja, $L = L_{\text{máximo}}$.
- O fator de redução de malha (θ), influencia pouco o tempo de CPU quando aplicado a malhas estruturadas triangulares. Tanto para o problema linear, quanto para o problema senoidal, o valor ótimo é $\theta_{\text{ótimo}} = 0,0625$. Na literatura, recomenda-se usar $\theta = 0,25$ para problemas em malhas estruturadas.

- O valor empregado para o fator de redução de malha influencia no número máximo de níveis do AMG para malhas triangulares. Quanto maior o valor empregado para θ , mais níveis são construídos.
- O fator de forte dependência na malha grossa (ε) influencia pouco o tempo de CPU quando aplicado a malhas estruturadas triangulares. Tanto para o problema linear, quanto senoidal, o valor que obteve o melhor desempenho médio é $\varepsilon = 0,35$, que está de acordo com a literatura.
- O efeito do número de incógnitas no tempo de CPU teve o comportamento esperado, crescendo quase linearmente com o número de incógnitas.
- O algoritmo otimizado gerou ganhos de pelo menos 16,0% quando comparado ao algoritmo padrão, além de ser resolvido com menos iterações externas (ciclos V).

6 Conclusão

Este capítulo apresenta as principais constatações deste trabalho bem como as contribuições desta tese. O objetivo deste trabalho era o de analisar qualitativamente os parâmetros do método *multigrid* algébrico em malhas estruturadas quadrangulares e triangulares. Com esta análise de parâmetros chegou-se a um algoritmo ótimo que proporcionou a minimização do tempo de CPU para as equações que foram aqui resolvidas.

Em um domínio de cálculo quadrado, foram discretizadas a equação de Laplace, considerando duas condições de contorno distintas (problema linear e senoidal), e a equação de Poisson. A equação de Poisson foi discretizada em malhas quadrangulares e a equação de Laplace foi discretizada em malhas triangulares e quadrangulares. Para malhas quadrangulares foi aplicado o método de diferenças finitas e para malhas triangulares, o método dos volumes finitos. Alguns dos resultados obtidos puderam ser comparados com os resultados disponíveis para o método *multigrid* geométrico, tanto em malhas quadrangulares, quanto em malhas triangulares.

O código computacional para o AMG foi baseado no código AMG1R6 de Ruge e Stüben (1986), sendo que foi considerada a fase *setup* e alterada a fase de solução. A forma de entrada dos dados também foi modificada, de forma a facilitar as alterações que deveriam ser feitas a cada simulação. O gerador para malhas triangulares foi obtido de Araki e Alves (2009). Para a verificação do código computacional foram feitos testes computacionais, onde foram escolhidas variáveis de interesse para que fossem analisadas com a utilização do programa Richardson_3p1, de autoria do orientador desta tese. Foram analisadas as ordens efetiva e aparente do erro, além do comportamento do erro numérico com o refinamento da malha, onde se concluiu que os códigos computacionais aplicados para a obtenção dos resultados numéricos estão coerentes.

Foram obtidos os parâmetros ótimos ou que obtiveram o melhor desempenho médio para as equações de Laplace e Poisson em malhas quadrangulares e para a equação de Laplace em malhas triangulares. Estes resultados puderam ser comparados entre si, com os

encontrados na literatura e ainda com os disponíveis para o GMG. Com base nos resultados obtidos pode-se concluir que:

- O número de iterações internas (ν) afeta significativamente o tempo de CPU do AMG. Para a equação de Laplace e Poisson discretizadas em malhas quadrangulares, o número ótimo (aquele que resulta no menor tempo de CPU) é 2. Para a equação de Laplace discretizada em malhas triangulares, o valor ótimo é 1. Na literatura, emprega-se $\nu = 1$ para o AMG em malhas quadrangulares.
- O número de níveis (L) empregados para a resolução do AMG também afeta significativamente o tempo de CPU. Independente da equação a ser resolvida ou do tipo de malha empregada na discretização do domínio de cálculo, observou-se que o número ótimo de níveis é o número máximo que pode ser empregado para cada tamanho de problema.
- O fator de redução de malha (θ), influencia pouco o tempo de CPU, quando aplicado a malhas estruturadas quadrangulares. Este parâmetro variou com o tipo de problema resolvido. Para o problema linear, $\theta = 0,40$, para o problema senoidal, $\theta = 0,25$, e para Poisson, $\theta = 0,20$. Para malhas triangulares, o valor ótimo é $\theta = 0,0625$. Na literatura, recomenda-se usar $\theta = 0,25$ para problemas em malhas estruturadas.
- O fator de forte dependência na malha grossa (ε) influencia pouco o tempo de CPU, quando aplicado a malhas estruturadas quadrangulares. Este parâmetro variou com o problema resolvido: tanto para o problema linear, quanto senoidal, $\varepsilon = 0,35$, e para Poisson, $\varepsilon = 0,20$. Para malhas triangulares, o valor ótimo é $\varepsilon = 0,35$. Na literatura, recomenda-se usar $\varepsilon = 0,35$ para problemas em malhas estruturadas.
- O número ótimo de iterações internas para o AMG em malhas quadrangulares coincide com o ótimo para o GMG para as equações de Laplace (problemas linear e senoidal) e de Poisson.
- O número ótimo de níveis para o AMG coincide com o número ótimo de níveis para o *multigrid* geométrico, para as equações de Laplace e Poisson, em malhas estruturadas quadrangulares.
- O método *multigrid* algébrico faz menos ciclos para atingir a tolerância quando comparado ao método *multigrid* geométrico, para os resultados obtidos em malhas quadrangulares.

- Para malhas quadrangulares, o GMG resolve o problema em 20% do tempo gasto pelo AMG.

As contribuições desta tese:

- Verificou-se que existe um número ótimo de iterações internas para o AMG aplicado às equações de Laplace e Poisson, tanto em malhas quadrangulares quanto em malhas triangulares. Estes valores diferem quanto ao tipo de malha empregada na discretização.
- Verificou-se que existe um número ótimo de níveis quando o AMG é aplicado às equações de Laplace e Poisson, tanto para malhas quadrangulares, quanto para malhas triangulares. Este valor coincide para os dois tipos de problema, quanto para os dois tipos de malha.
- Constatou-se que, para malhas quadrangulares, o fator de redução de malha mostrou-se dependente do problema. Para malhas triangulares, não foi observado este comportamento, sendo que o valor ótimo obtido é o menor testado (corresponde ao engrossamento levando em conta mais pontos). Além disso, é o valor que corresponde à resolução do problema com o menor número de malhas auxiliares. Pode-se dizer que o comportamento para este parâmetro depende do tipo de malha que está sendo empregada na discretização.
- Observou-se que, para a equação de Laplace (independente do tipo de malha empregado) existe um valor ótimo para o fator de forte dependência na malha grossa. Para a equação de Poisson, o valor ótimo é diferente.
- Constatou-se que os parâmetros ótimos obtidos (número de iterações internas e número de níveis) para as equações de Laplace e Poisson, discretizadas em malhas quadrangulares, coincidem com os valores ótimos obtidos para o *multigrid* geométrico aplicado à equação de Laplace.
- Foram obtidos resultados em malhas mais refinadas que as apresentadas na literatura para malhas triangulares.

Recomendações práticas:

Com base nos resultados desta tese, podem ser feitas algumas recomendações práticas aos usuários do método *multigrid* algébrico para problemas iguais ou similares aos estudados.

Problemas em malhas quadrangulares estruturadas:

- Número de iterações internas: $\nu = 2$;
- Número de níveis de malha: $L = L_{\text{máximo}}$;
- Fator de redução de malha: equação de Laplace com solução linear: $\theta = 0,40$; equação de Laplace com solução senoidal: $\theta = 0,25$ e equação de Poisson: $\theta = 0,20$.
- Fator de forte dependência na malha grossa, para a equação de Laplace: $\varepsilon = 0,35$.
- Fator de forte dependência na malha grossa, para a equação de Poisson: $\varepsilon = 0,20$.

Problemas em malhas triangulares estruturadas:

- Número de iterações internas: $\nu = 1$;
- Número de níveis de malha: $L = L_{\text{máximo}}$;
- Fator de redução de malha: $\theta = 0,0625$;
- Fator de forte dependência na malha grossa: $\varepsilon = 0,35$.

Propostas para trabalhos futuros:

As sugestões para trabalhos futuros, que puderam ser identificadas com a realização deste trabalho estão listadas abaixo:

- **Algoritmo FAS:** aplicar o algoritmo FAS às equações aqui estudadas, sendo que estes resultados podem ser comparados aos aqui apresentados.
- **Estudo de parâmetros:** outros testes computacionais podem ser feitos com estas equações. Os efeitos da variação do *solver*, da estimativa inicial, tipos de restrição, tipos de prolongação, critério de parada, por exemplo, podem ser avaliados.
- **Aspectos qualitativos para outras equações:** outras equações de Dinâmica de Fluidos podem ser resolvidas com o uso do AMG. Por exemplo, a equação de advecção difusão, equação de Burgers, que envolve a resolução de um sistema de equações. Estas equações poderiam ser resolvidas tanto em malhas quadrangulares

quanto em malhas triangulares, onde seriam obtidos os parâmetros ótimos para cada uma destas equações.

- **Malhas não-estruturadas:** as equações resolvidas neste trabalho podem ter seus parâmetros ótimos obtidos em malhas não-estruturadas.
- **Discretizações em 3D:** resolver as mesmas equações abordadas nesta tese, discretizadas em domínios tridimensionais, realizando um estudo similar a respeito dos parâmetros ótimos.

Referências

- ARAKI, L. K., ALVES, A. Comunicação pessoal a Roberta Suero, UFPR, 2009.
- BRANDT, A. “Multi-Level Adaptative Solutions to Boundary-Value Problems”, *Mathematics and Computation*, v. 31, p. 333-390, 1977.
- BRANDT, A. “Algebraic *Multigrid* Theory: The Symmetric Case”, 1986.
- BRIGGS, W. L., HENSON, V. E., MCCORMICK, S. F. “A *multigrid* tutorial”, 2ª ed., SIAM, 2000.
- BURDEN, R. L., FAIRES, J. D. “Análise Numérica”, Thomson, 2009.
- CAMPOS, F. O., OLIVEIRA, R. S., SANTOS, R. W. “Performance Comparison of parallel geometric and algebraic *multigrid* preconditioners for the bidomain equations”, Springer-Verlag Berlin Heidelberg, 2006.
- CHANG, Q., WONG, Y. S., FU, H. “On the Algebraic *Multigrid* Method”, *Journal of Computation Physics*, v. 125, p. 279-292, 1996.
- CLEARY, A. J., FALGOUT, R. D., HENSON, V. E., JONES, J. E., MANTEUFFEL, T. A., MCCORMICK, S. F., MIRANDA, G. N., RUGE, J. W., “Robustness and Scalability of Algebraic *Multigrid*”, *SIAM Journal and Scientific Computing*, v. 21, pp. 1886-1908, 2000.
- CRAIG, C. D. 12122_tex2html_wrap26.gif. 1995. Altura: 587 pixels. Largura: 372 pixels. 6.15 Kb. Formato GIF Image.
Disponível em: < <http://www.mgnet.org/mgnet/tutorials/xwb/mg.html> >. Acesso em 20 set. 2007.
- CORDAZZO, J. “Simulação de reservatórios de petróleo utilizando o método EbFVM e *multigrid* algébrico”, Tese de doutorado em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, 2006.
- CUNHA, M. C. “Métodos Numéricos”, Editora Unicamp, 2003.
- DIETRICH, B. B. “Towards Algebraic *Multigrid* for Elliptic Problems of Second Order”, *Computing*, v. 55, p. 379-393, 1995.
- DUMETT, M. A., VASSILEVSKI, P., WOODWARD, C. S. “A *Multigrid* Method for Nonlinear Unstructured Finite Element Elliptic Equations”, *SIAM Journal and Scientific Computing*, 2002.

FALGOUT, R. D. “An Introduction to Algebraic *Multigrid*”, Computing in Science and Engineering, 2006.

FEDORENKO, R. P., “On the Speed of Convergence of an Iteration Process”, USSR Comput. Math. And Math. Phys., v. 4 (3), 1964.

FERZIGER, J. H., PERIC, M. “Computational Methods for Fluid Dynamics”, Berlin: Springer-Verlag, 2002.

FORTUNA, A. O. “Técnicas Computacionais para Dinâmica de Fluidos”, Edusp, 2000.

GASPAR, F. J., GRACIA, J. L., LISBONA, F. J., RODRIGO, C. “On geometric *multigrid* methods for triangular grids using three-coarsening strategy”, Applied Numerical Mathematics, 2009.

GREENBERG, M. D. “Advanced Engineering Mathematics”, 2ª Edição, Prentice Hall, 1998.

HAASE, G., LANGER, U. “*Multigrid* Methods: From Geometrical to Algebraic Versions”, em Bourlioux A. e Gander, M. J., Modern Methods in Scientific Computing and Applications, Kluwer Academic Press, Dordrecht, pp. 103-154, 2002.

HACKBUSCH, W. “Multi-grid Methods and Applications”, Springer-Verlag, Berlim, 1985.

HUGHES, T. J. R. “The Finite Element Method – Linear Static and Dynamics Finite Element Analysis”, Mineola (USA): Dover Publications, Inc., 2000.

IWAMURA, C., COSTA, F.S., SBARSKI, I., EASTON, A., LI, N. “An efficient algebraic *multigrid* preconditioned conjugate gradient solver”, *Computer Methods in Applied Mechanics and engineering*, vol. 192, pp. 2299-2318, 2003.

KRECHEL, A., STÜBEN, K. “Operator Dependent Interpolation in Algebraic *Multigrid*”, Proceedings of Fifth European *Multigrid* Conference, 1999.

KREYSZIG, E. “Advanced Engineering Mathematics”, 8ª Edição, Wiley, 1999.

LANGER, U., PUSCH, D. “Comparison of Geometric and Algebraic *Multigrid* Preconditioners for data-sparse boundary element matrices”, *Springer-Verlag Berlin Heidelberg*, 2006.

MALISKA, C. R. “Transferência de Calor e Mecânica dos Fluidos Computacional”, LTC, 2 ed., 2004.

MARCHI, C. H. “Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos”, Tese de Doutorado em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, 2001.

MINKOWYCZ, W. J., SPARROW, E. M. Schneider, G. E., Pletcher, R. H. “Handbook of Numerical Heat Transfer”, John Wiley & Sons, Inc., 1988.

OBERKAMPF, W. L., BLOTTNER, F. G. "Issues in computational fluid dynamics code verification and validation", *AIAA Journal*, vol. 36, n.5, pp. 687-695, 1998.

OLIVEIRA, F., PINTO, M. A. V., MARCHI, C. H. "Efeito de Roteiros do Método *Multigrid* sobre o Tempo de CPU para a Equação de Laplace 2D", Proceedings of CILAMCE, 2008.

PEREIRA, F. H., VERARDI, S. L. L., NABETA, S. I. "A fast algebraic *multigrid* preconditioned conjugate gradient solver", *Applied Mathematics and Computational*, v. 179, p. 344-351, 2006.

PEREIRA, F. H. O método *multigrid* algébrico na resolução de sistemas lineares oriundos do método dos elementos finitos, Tese de Doutorado em Engenharia Elétrica, Universidade de São Paulo, São Paulo, SP, Brasil, 2007.

REDDYE, J. N., GARTLING, D. K. "The finite Element Method in Heat Transfer and Fluid Dynamics", Boca Raton (USA): CRC Press, 1994.

REITZINGER, S., SCHREIBER, U., RIENEN, U. "Algebraic *multigrid* for complex symmetric matrices and applications", *Journal of Computational and Applied Mathematics*, v. 155, p. 405-421, 2003.

ROACHE, P. J. "Verification and Validation in Computational Science and Engineering", Albuquerque, USA: Hermosa, 1998.

RUGE, J., STÜBEN, K. 'Algebraic *Multigrid* (AMG)'. Em S. F. McCormick, *Multigrid Methods*, volume 5 de *Frontiers in Applied Mathematics*, p. 1-57, SIAM, Philadelphia, 1986.

SCHNEIDER, G. E., ZEDAN, M. "A modified Strongly Implicit Procedure for Numerical Solution of Field Problems", *Numerical Heat Transfer*, v. 4, p. 1-19, 1981.

STÜBEN, K. "A review of algebraic *multigrid*", *Journal of Computational and Applied Mathematics*, v. 128, p. 281-309, 2001.

SZABÓ, B.; BABUSKA, I. "Finite element analysis", New York: Wiley, 1991.

TANNEHILL, J. C., ANDERSON, D. A., PLETCHER, R. H. "Computational Fluid Mechanics and Heat Transfer", Taylor & Francis, 1997.

TROTTEBERG, U., OOSTERLEE, C., SCHÜLLER, A. "*Multigrid*", Academic Press, 2001.

VERSTEEG, H. K., MALALASEKERA, W. "An introduction to Computational Fluid Dynamics: The Finite Volume Method", 2 ed. Harlow, England: Pearson/Prentice Hall, 2007.

WATANABE, K., IGARASHI, H., HONMA, T. "Comparison of Geometric and Algebraic *Multigrid* Methods in Edge-Based Finite-Element Analysis", *IEEE Transactions on Magnetics*, v. 41, 2005.

WESSELING, P., "An introduction to *Multigrid* Methods", John Wiley & Sons, 1992.

WU, C. T., ELMAN, H. C. “Analysis and comparison of geometric and algebraic *multigrid* for convection-diffusion equations”, SIAM Journal on Scientific Computing, v. 28, n° 6, p. 2208-2228, 2006.

Apêndice A – Definição do operador de interpolação

Para a determinação do operador de interpolação, deve-se partir da definição de erro algebricamente suave (BRGGS et al., 2000 e PEREIRA, 2007). Tem-se que o erro a ser aproximado em Ω^h é tal que:

$$A_h e_h \approx 0 \quad (\text{A.1})$$

que pode ser reescrito da forma:

$$a_{ii} e_i \approx - \sum_{j \in N_i} a_{ij} e_j \quad \text{e } j \neq i \quad (\text{A.2})$$

Particionando N_i em C_i , D_i^s e D_i^w tem-se que:

$$a_{ii} e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} e_j - \sum_{j \in D_i^w} a_{ij} e_j \quad (\text{A.3})$$

Faz-se necessário substituir e_j no segundo e terceiro somatórios, com aproximações em termos de e_i ou e_j . Se $j \in D_i^w$, então e_i não depende fortemente de e_j e, conseqüentemente, a_{ij} é pequeno. Assim, introduzindo um erro relativamente pequeno, pode-se fazer:

$$\sum_{j \in D_i^w} a_{ij} e_j \approx \sum_{j \in D_i^w} a_{ij} e_i \quad (\text{A.4})$$

Logo, a Eq. (A.3), torna-se:

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} e_j \quad (\text{A.5})$$

Para os pontos $j \in D_i^s$ a questão é um pouco mais complicada por se tratar de pontos que mantêm forte influência sobre i . Entretanto, $j \in D_i^s$ significa que j depende fortemente de (no mínimo) um ponto em C_i . Portanto, é razoável aproximar e_j , com $j \in D_i^s$, por uma média ponderada dos pontos em C_i da seguinte forma:

$$e_j \approx \frac{\sum_{k \in C_i} a_{jk} e_k}{\sum_{k \in C_i} a_{jk}} \quad (\text{A.6})$$

Substituindo a Eq. (A.6) em (A.5), chega-se a:

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i \approx - \left[\sum_{j \in C_i} a_{ij} e_j + \sum_{j \in D_i^s} a_{ij} \left(\frac{\sum_{k \in C_i} a_{jk} e_k}{\sum_{k \in C_i} a_{jk}} \right) \right] \quad (\text{A.7})$$

Rearranjando os termos, chega-se a:

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i = - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} \left(\frac{\sum_{k \in C_i} a_{jk} e_k}{\sum_{k \in C_i} a_{jk}} \right) \quad (\text{A.8})$$

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i = - \sum_{j \in C_i} a_{ij} e_j - \frac{\sum_{j \in D_i^s} \sum_{k \in C_i} a_{ij} a_{jk} e_k}{\sum_{k \in C_i} a_{jk}} \quad (\text{A.9})$$

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i = - \sum_{j \in C_i} a_{ij} e_j - \frac{\sum_{m \in D_i^s} \sum_{k \in C_i} a_{im} a_{mk} e_k}{\sum_{k \in C_i} a_{mk}} \quad (\text{A.10})$$

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i = - \sum_{j \in C_i} \left[a_{ij} + \sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right) \right] e_j \quad (\text{A.11})$$

$$e_i = - \sum_{j \in C_i} \frac{a_{ij} + \sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in D_i^w} a_{in}} e_j \quad (\text{A.12})$$

Usando o fato de que $e_j^h = e_j^{2h}$ para $j \in C$, obtêm-se a seguinte expressão:

$$e_i = - \sum_{j \in C_i} \frac{a_{ij} + \sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in D_i^w} a_{in}} e_j \quad (\text{A.13})$$

onde

$$w_{ij} = - \frac{a_{ij} + \sum_{m \in D_i^s} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in D_i^w} a_{in}} \quad (\text{A.14})$$

Apêndice B – Estudo complementar para o fator de redução de malha

O fator de redução de malha é uma constante que quantifica o quanto um ponto está fortemente conectado a outros. Seu valor influencia diretamente na geração das malhas auxiliares do AMG (malhas com menos incógnitas). Para a seleção dos valores a serem aqui testados são considerados dois motivos: valores que são empregados na resolução de problemas com o uso do AMG e valores que têm relação com as razões de engrossamento para o método *multigrid* geométrico.

O estudo do fator de redução de malha (θ) para o problema linear (equação de Laplace), discretizado em malhas quadrangulares, revelou como sendo o valor ótimo para este parâmetro $\theta = 0,4$. Este é o valor máximo testado em todos os casos considerados nesta tese. Com isso, sentiu-se a necessidade de se testar valores maiores para θ , com o objetivo de se verificar se, valores maiores que $\theta = 0,4$ resultariam em tempos de CPU menores. Para tanto, são realizadas algumas simulações para valores maiores. São testados $\theta = 0,45$, $\theta = 0,5$ e $\theta = 0,55$. Na Tab. B.1 tem-se as diferenças entre os tempos de CPU para os valores testados para θ :

Tabela B.0.1: Diferenças entre os tempos de CPU para o fator de redução de malha para o problema linear em malhas quadrangulares

θ	257x257	1025x1025	4097x4097
0,0625	10%	3,8%	1,6%
0,11	3,3%	0,2%	4,6%
0,2	6,7%	4,2%	5,7%
0,25	-----	1,6%	5,5%
0,4	-----	-----	-----
0,45	-----	3,5%	2,3%
0,5	3,3%	1,6%	5,5%
0,55	*	*	*

Na Tab. B.1 o símbolo “*” significa porcentagens muito grandes, acima de 100%. Conforme pode ser observado nesta tabela, valores maiores que $\theta = 0,4$ não resultam em menores tempos de CPU para o problema linear, resolvido em malhas quadrangulares.